intel

APPLICATION NOTE

AP-137

September 1981

8298 Functional Specification and Firmware Description

Kevin Corbett and Randy Battat
Special Products Division
Applications Engineering

- **Supervises Writing, Erasing and Reading of Intel E^2PROMs**
- **Supports Up to 16K Bytes of E^2 Memory**
- **Provides Optimized Read/Write Commands for Efficient Data Transfers**
- **Microprocessor Peripheral, Compatible with iAPX 86/88, MCS®-80/85, MCS®-48 Families**
- **Provides Data Check to Minimize Necessary E^2 Writes**
- **Supports Interrupt or Polled Operation**

The 8298 Intelligent E^2PROM (Electrically Erasable Programmable Read Only Memory) Controller is a microprocessor peripheral designed to efficiently oversee reading, writing, and erasing of Intel E^2 devices. Up to 16K bytes of E^2PROM memory are supported by the 8298 with a few external components. These components include an 8243 I/O expander which provides address and control signals, a V_{PP} Switch to provide program pulses, and an $\overline{OE}$ switch to provide for erasure of a complete chip. The 8298 also provides local bus request and acknowledge signals enabling the E^2PROMs to couple directly with system busses.

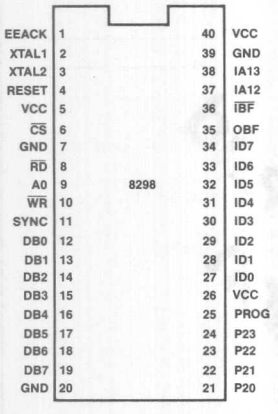


Figure 1. Pinouts of 8298

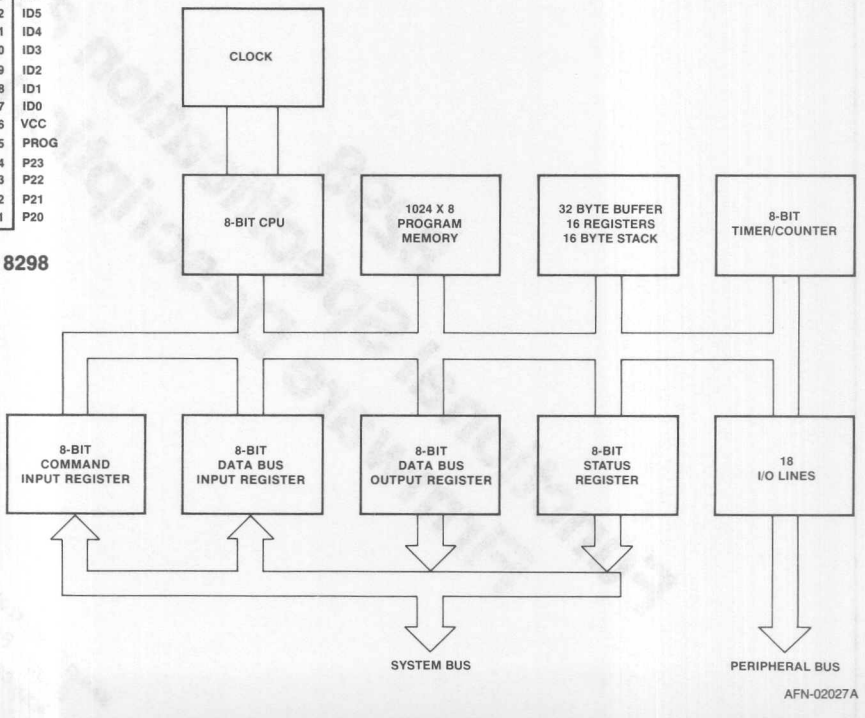


Figure 2. 8298 Block Diagram

Table 1. Pin Description (8298)

Symbol	Pin No.	Type	Function
EEACK	1	I	E^2 Acknowledge—a logical 1 on this pin indicates that the 8298 has access to the E^2PROM memory (via local bus).
XTAL1, XTAL2	2, 3	I	Inputs for a crystal, LC circuit, or external timing signal to determine internal oscillator frequency.
RESET	4	I	Used to initialize the chip to a known state during power on.
$\overline{CS}$	6	I	Chip Select Input—used to select the 8298 for other devices on the common data bus.
$\overline{RD}$	8	I	I/O read input which allows the master CPU to read from the 8298.
A0	9	I	Address Line—used to select between data and status registers during read operations and to distinguish between data and commands written to the 8298 during write operations.
$\overline{WR}$	10	I	I/O or Memory Write input which allows the master CPU to write the 8298.
SYNC	11	O	8298 cycle synchronization signal.
DB0–DB7	12–19	I/O	8 bidirectional lines used for communications between the central processor and the 8298 registers.
GND	7, 20, 39	P.S.	Circuit ground potential.
P20–P23	21–24	I/O	I/O information passed to the 8243 I/O expander.
PROG	25	O	Latching signal for 8243 I/O expander.
ID0–UD7	27–34	I/O	Internal data bus, 8 bidirectional data lines common to the 8298 and all E^2 devices.
OBF	35	O	Output Buffer Full—used to interrupt the host CPU when the data output buffer of the 8298 is full.
$\overline{TBF}$	36	O	Input Buffer Full—used to interrupt the host CPU when the 8298 has emptied the input buffer.
IA12, IA13	27, 38	O	Internal address bits 12, 13.
V_{CC}	5, 40, 26	P.S.	+5V supply input ±10%.
IA0–IA11	1–5, 17–23	I/O	Internal address bus—12 bidirectional address lines common to the 8298 and all E^2 devices.
P20–P23	8–11	I/O	I/O information passed by 8298.
PROG	7	O	I/O information latching signal.
$\overline{V_{PP}}$ EN	13	O	V_{PP} control—activates programming voltage to V_{PP} switch when low.
$\overline{CRD}$	14	O	Controller Read—active low enables E^2PROM output buffers when 8298 performs an E^2 read operation.
$\overline{EEN}$	15	O	E^2 Enable—active low enables E^2PROM devices.
$\overline{EEREQ}$	16	O	E^2 Request—active low requests 8298 access to E^2 devices. 8298 will not take control of E^2PROMs (via EEN) until EEACK is brought high.
V_{CC}	24	P.S.	±5V power supply, ±10%.
GND	12	P.S.	System ground potential.

GENERAL DESCRIPTION

E^2 Operation

The 8298 receives commands and data from the host CPU to perform E^2 write, erase, and read operations. (See Figure 2 for a block diagram of the 8298.)

Before any E^2 operation is performed, however, the 8298 requests access to the local E^2 memory bus by bringing EEREQ low. If the E^2 memory bank is not being accessed by another processor, the external hardware will raise EEACK high. Systems in which the 8298 is the only device connected to E^2 memory may tie EEACK high so the 8298 is always granted access.

The 8298 performs a read operation by outputting the address of the selected location and bringing $\overline{EEN}$ (E^2 Enable) and $\overline{CRD}$ (Controller Read) low. Data is then read through the internal data bus I/O lines of the 8298 and all control signals are returned to their inactive state. Figure 3 shows a general system interface diagram for reference.

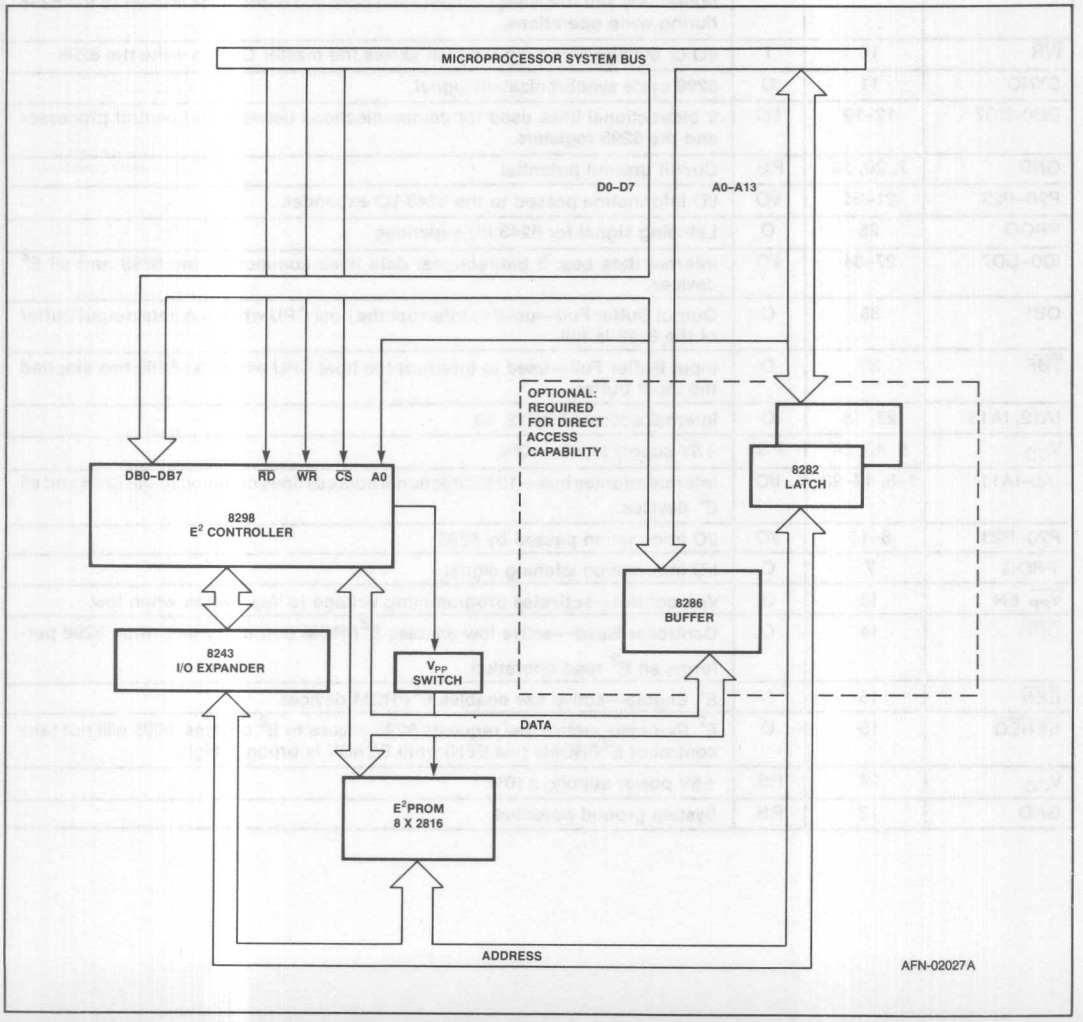


Figure 3. System Interfaces Diagram

Write and erase operations are performed in a similar manner. Address and data are output and $\overline{V_{PP}EN}$ is pulsed low for the duration of the write/erase cycle. V_{PP} pulse width is dependent upon clock oscillator frequency and is software programmable. The default value is 10 msec for a clock frequency of 6 MHz. At the end of the write/erase cycle, $\overline{V_{PP}EN}$ is brought inactive (high) and the 8298 delays while the 2816 programming voltage (V_{PP}) falls to 5V. This delay is guaranteed to be greater than 100 μsec at 6 MHz clock frequency; at lower frequencies the delay is longer.

When the 8298 receives a write command, the E^2 location to be written is first read. If the data is the same as the data currently in memory, the write operation is terminated. If not, the 8298 checks to see if a byte erase operation is necessary. An erase is necessary when the data to write contains a logical 1 where there is a logical 0 already stored. A byte erase operation is performed by writing all 1's to the selected byte. Following the optional erase cycle, the write cycle is performed.

HOST CPU INTERACTION

The host CPU and 8298 communicate with each other by means of four registers; two input and two output registers. Commands are issued when the host CPU writes a byte to the 8298 command register. Additional information required by the 8298 is transmitted by the host writing data bytes to the 8298 data input register. The 8298 transmits data back to the host (e.g., data read from E^2PROM) via its data output register. Data transfer is synchronized by interrupt lines or the status register which can be read by the host at any time. Register selection is done by the $\overline{RD}$ and $\overline{WR}$ signals and by A0 as shown in Figure 4.

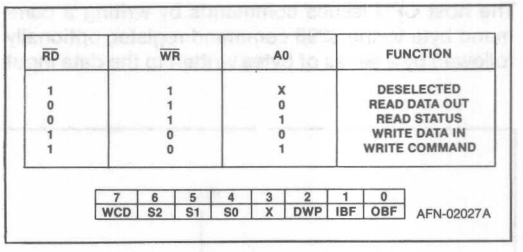


Figure 4. 8298 Register Selection

The status register can be read at any time to determine the state of the 8298. It is defined as follows:

OBF — Output Buffer Full—The data output register has data available for the host to read.

IBF — Input Buffer Full—The data input register or command register contains information not yet recognized by the 8298. The host CPU should never write to the 8298 when IBF = 0.

DWP — Direct Write Possible—The host CPU may perform a "Direct Write" if this bit is set and all other bits indicate that the UPI is waiting for a command.

WCD — Waiting for Command/Data—The 8298 is waiting for the host CPU to write a byte to its command or data input register.

S0, S1, S2 — Additional status information can be read at any time to determine the state of the 8298. It is defined as follows:

WCD	S2	S1	S0	OBF	Function
1	1	1	1	0	Waiting for command
0	0	0	0	0	Executing command
1	1	1	0	0	Illegal command—waiting for new command
0	0	X	X	1*	Illegal commmand transmitted during E^2 write cycle
1	0	X	X	1*	Command issued data expected
1	1	1	1	1	Read command complete—data output register contains data read
0	0	0	0	1	Series read command data available
0	1	0	1	0	Write cycle in progress

WCD	S2	S1	S0	OBF	Function
1	0	0	0	0	Waiting for data byte #1
1	0	0	1	0	Waiting for data byte #2
1	0	1	0	0	Waiting for data byte #3
1	9	1	1	0	Waiting for data byte #4

X = Don't care.
* = Data output register contains 10101010 to indicate illegal operation.

COMMANDS

The host CPU issues commands by writing a command byte to the 8298 command register, optionally followed by a series of bytes written to the data input register. The status register indicates the type of data expected and when the 8298 is ready for commands and data. Figure 5 shows a flowchart detailing the transmission of a command.

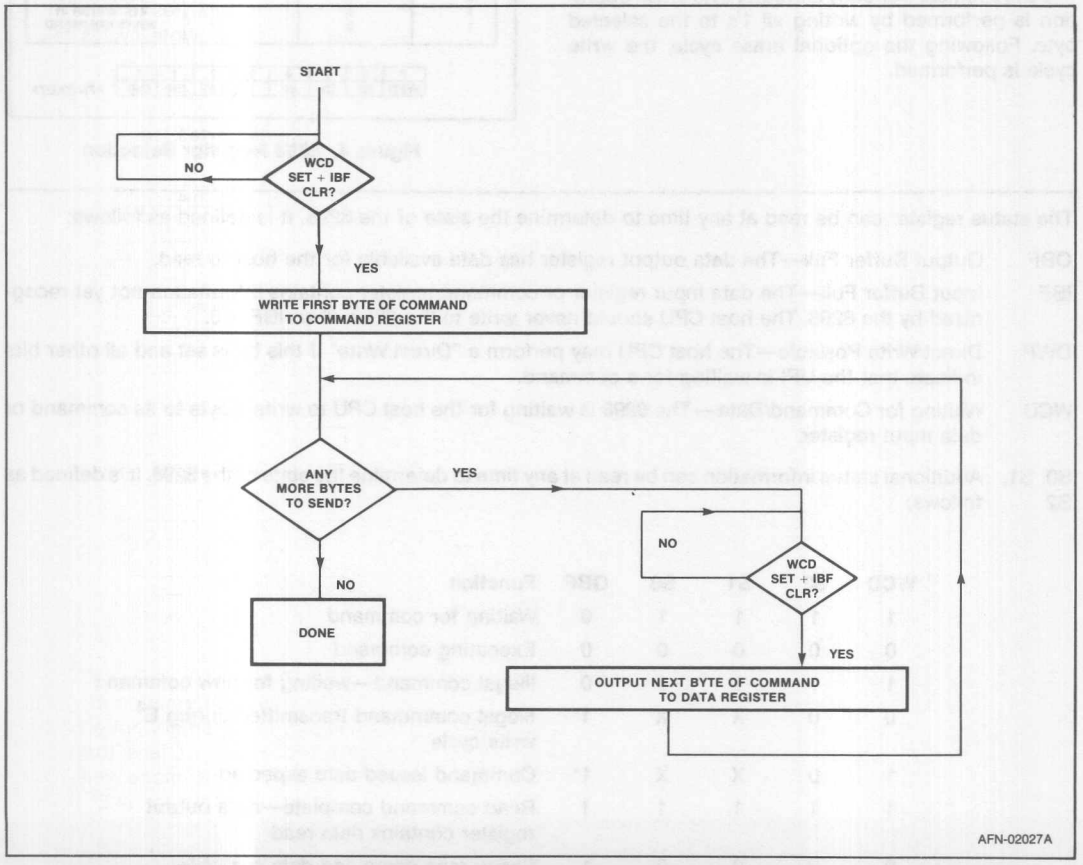


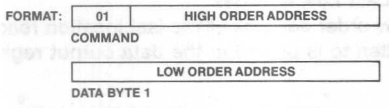
Figure 5. Flow Chart for Command Transmission

READ COMMANDS

The following commands are issued to read data from E^2 memory.

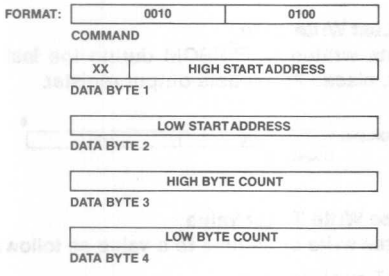
Indirect Read

Read the contents of a single location. Data read is passed to the host via the data output register. The command consists of the following sequence.



Series Read

Read up to 16,384 sequential locations. Data read is placed in the data output register. As soon as the host reads a byte from this register, it is reloaded with data read from the next location. Note that the host CPU should determine that data is available by verifying that the OBF bit is set in the status register.

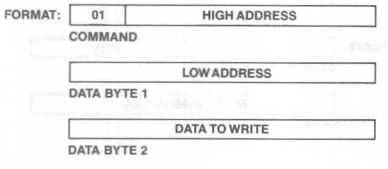


WRITE/ERASE COMMANDS

Five different write and erase commands are provided.

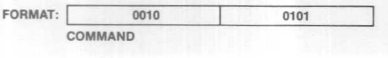
Indirect Write

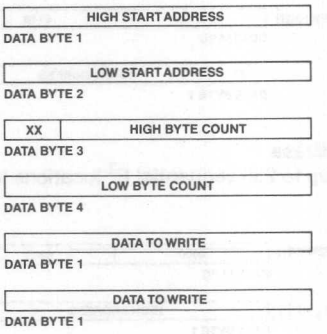
Write a single E^2 location



Series Write

Write up to 16,384 sequential locations with increasing addresses.

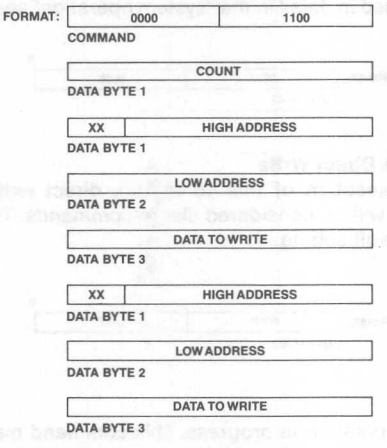




Note that the 8298 expects "data byte 1" for each byte of data transmitted. Also, the 8298 internally buffers up to 32 data bytes at a time. Therefore, series writes of less than 33 locations are performed efficiently because the host CPU outputs all bytes to write in a "burst" manner before actual E^2 write cycles take place.

Multiple Write

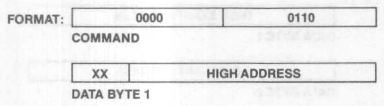
Write up to 256 non-sequential bytes.



Note the sequence (data byte 1, 2, 3, 1, 2, 3 . . .) of data in which the 8298 receives information. Up to 9 address/data combinations are internally buffered by the 8298. Thus multiple writes of less than 10 locations are performed in a "burst" mode where all bytes are buffered by the 8298 before actual E^2 write cycles take place.

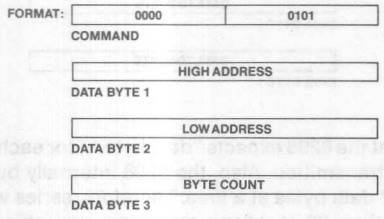
Chip Erase

Erase an entire E^2PROM. The high order address is required to select the device to be erased.



Block Erase

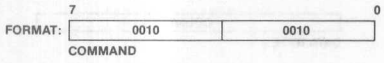
Erase up to 256 sequential E^2 locations in increasing order.



UTILITY COMMANDS

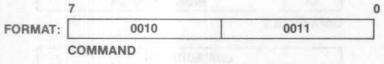
Enable Direct Write

Enables direct writes. A direct write is performed when the 8298 received data when it is waiting for a command and direct writes are enabled. This topic is discussed in detail in the "system operation" section.



Disable Direct Write

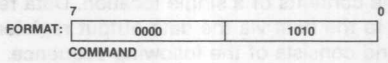
After execution of this command, direct write attempts will be considered illegal commands. This is the default setting.



Abort

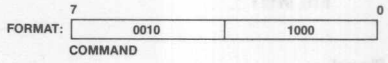
Abort operation in progress. This command may be given at any time. If the 8298 is in the middle of a write cycle, writing is stopped, V_{PP} is brought low, and after a delay while V_{PP} falls to 5V (approximately 100 μsec at 6 MHz), the 8298 waits for another command.

After issuing an abort command, the 8298 may go back and, using the commands described below, determine the address and data of the interrupted write operation.



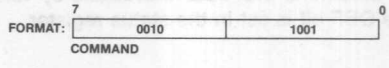
Read Last Low Address

The low order address of the last location read from or written to is placed in the data output register.



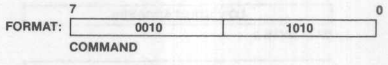
Read Last High Address

The high order address of the last location read from or written to is placed in the data output register.



Read Last Write Data

The data written to E^2PROM during the last write cycle is placed in the data output register.



Initialize Write Timer Value

Initializes write cycle time to a value as follows:

$$\text{Write Timer Value}_{10} = 256 - \frac{(\text{Clock Frequency (Hz)} * \text{Write Time (Sec)})}{480}$$

Thus, for a 10 Msec write time with a 6 MHz clock, the write timer value should be 131_{10}. This is the default value.

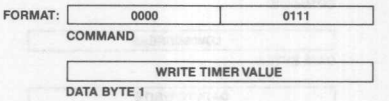


Table 2

Command	# of Bytes for Commands	Byte#	Format	Next Status
INDIRECT READ	2	1 2	01 High Address Low Address	DB1 OBF
SERIES READ	5	1 2 3 4 5	0010 0100 XX High Start Address Low Start Address High Byte Count Low Byte Count	DB1 DB2 DB3 DB4 OBF
INDIRECT WRITE	2+ WRITE DATA	1 2 3	01 High Address Low Address DATA TO WRITE	DB1 DB2 WCD
SERIES WRITE	5+ WRITE DATA	1 2 3 4 5	0010 0101 XX High Start Address Low Start High Byte Count Low Byte Count DATA TO WRITE	DB1 DB2 DB3 DB4 DB1 DB1/WCD NOTE 1
MULTIPLE WRITE	2+ ADDRESS/WRITE DATA	1 2 * * *	0000 1100 Byte Count XX High Address Low Address DATA TO WRITE	DB1 DB1 DB2 DB3 DB1/WCD NOTE 2
BLOCK ERASE	4	1 2 3 4	0000 0101 High Start Address Low Start Address Byte Count	DB1 DB2 DB3 WCD
CHIP ERASE	2	1 2	0000 0110 XX High Address	DB1 WCD NOTE 3
ENABLE DIRECT WRITE	1	1	0010 0010	WCD
DISABLE DIRECT WRITE	1	1	0010 0011	WCD
ABORT	1	1	0000 1010	WCD
READ LAST LOW ADDRESS	1	1	0010 1000	WCD
READ LAST HIGH ADDRESS	1	1	0010 1001	WCD
INITIALIZE WRITE TIMER VALUE	2	1 2	0000 0111 Write Timer Value	DB1 WCD

*ADDRESS + WRITE DATA

NOTES:

1. SERIES WRITE returns a status of DB1 whenever it is waiting for write data. When BYTE COUNT bytes have been sent, it will return a status of WCD. See SERIES WRITE command, under Indirect Configuration Commands, for more information.
2. MULTIPLE WRITE works in a loop format once the command and byte count are received. It requires groups of three bytes and cycles through the status as: -DB1—DB2—DB3- until "count" groups of bytes are received. It then returns a status of WCD.
3. High address is the high-order six bits of the highest address in an individual E^2PROM. An example might be: FOR 7FFH as the highest address in an E^2PROM, to erase would require sending 0000 0111 as the high address.

E^2PROM INTERFACE

The 8298 requires a few external components to drive E^2PROMs. These components are organized as functional blocks as follows:

V_{PP} Switch

Used to gate 21 V to V_{PP} line of E^2PROMs for writing and erasing. See AP 102.

OE Switching

Used to gate the +12V to E^2PROM OE line for the chip erase function. The OE switch is also responsible for pulling OE low during Read operation.

Direct Access Circuits

Required only if the 8298 is to support direct reading and writing of E^2PROM.

Direct reading is a function whereby the host CPU treats E^2 memory as EPROM memory and reads directly without the supervision of the 8298.

Direct writing is similar to direct reading in that the host CPU thinks it is writing to RAM. External circuitry forces a "write data register" operation to the 8298, so the data to write is latched in the 8298 data input register. External latches connected in parallel with 8298 Internal Address Lines latch the address. The 8298 reads the contents of these latches in order to determine the destination address of the write cycle. The E^2 controller seizes the internal busses, performs the write operation, and signals the host CPU when done.

The 8298 recognizes a direct write command when it is waiting for a command, *data* is received, and direct writing is enabled.

The advantage of the direct write command is that the host CPU can use its memory reference instructions to access E^2 memory.

Chip Erase Signal

The chip erase signal is used by the OE switch to gate +12V to the OE line of E^2PROMs during the chip erase command.

This signal is multiplexed with ID7 (internal data, bit 7) and should be latched on a high-to-low transition of EEN.

SYSTEM OPERATION

The 8298 can communicate with the rest of the system using the two mechanisms described below:

1. I/O Polling. The host CPU repeatedly reads the 8298 status register to determine when commands and data can be transmitted.

2. Interrupts: The 8298 WCD and OBF lines are used to interrupt the host processor when the 8298 is ready to accept input or has output available.

Reset and Power-Up Procedure

Reset is used to bring the 8298 into a known state upon power-up. It must be held low at least 10 msec after the power supply is within tolerance.

APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . – 65°C to + 150°C
Voltage on Any Pin With Respect to Ground . . . . . 0.5V to + 7V
Power Dissipation . . . . . 1.5 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

(T_A=0°C to 70°C, V_{SS}=0V: 8298; V_{CC}=V_{DD}=+5V±5%)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)	– 0.5	0.8	V	
V_{IL1}	Input Low Voltage (XTAL1, XTAL2, $\overline{RESET}$)	– 0.5	0.6	V	
V_{IH}	Input High Voltage (Except XTAL1, XTAL2, $\overline{RESET}$)	2.2	V_{CC}		
V_{IH1}	Input High Voltage (XTAL1, XTAL2, $\overline{RESET}$)	3.8	V_{CC}	V	
V_{OL}	Output Low Voltage (D_0-D_7)		0.45	V	I_{OL} = 2.0 mA
V_{OL1}	Output Low Voltage ($P_{10}P_{17}$, $P_{20}P_{27}$, Sync)		0.45	V	I_{OL} = 1.6 mA
V_{OL2}	Output Low Voltage (Prog)		0.45	V	I_{OL} = 1.0 mA
V_{OH}	Output High Voltage (D_0-D_7)	2.4		V	I_{OH} = – 400 μA
V_{OH1}	Output High Voltage (All Other Outputs)	2.4		V	I_{OH} = – 50 μA
I_{IL}	Input Leakage Current (T_0, T_1, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A_0, EA)		± 10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OZ}	Output Leakage Current (D_0-D_7, High Z State)		± 10	μA	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$
I_{LI}	Low Input Load Current ($P_{10}P_{17}$, $P_{20}P_{27}$)		0.5	mA	V_{IL} = 0.8V
I_{LI1}	Low Input Load Current ($\overline{RESET}$, $\overline{SS}$)		0.2	mA	V_{IL} = 0.8V
I_{DD}	V_{DD} Supply Current		15	mA	Typical = 5 mA
I_{CC} + I_{DD}	Total Supply Current		125	mA	Typical = 60 mA

A.C. CHARACTERISTICS

(T_A=0°C to 70°C, V_{SS}=0V: 8298; V_{CC}=V_{DD}=+5V ±5%)

REGISTER READ

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	$\overline{CS}$, A_0 Setup to $\overline{RD}$↓	0		ns	
t_{RA}	$\overline{CS}$, A_0 Hold After $\overline{RD}$↑	0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	250		ns	
t_{AD}	$\overline{CS}$, A_0 to Data Out Delay		225	ns	C_L = 150 pF
t_{RD}	$\overline{RD}$↓ to Data Out Delay		225	ns	C_L = 150 pF
t_{DF}	$\overline{RD}$↑ to Data Float Delay		100	ns	
t_{CY}	Cycle Time (Except 8298)	2.5	15	μs	6.0 MHz XTAL
t_{CY}	Cycle Time (8298)	4.17	15	μs	3.6 MHz XTAL

REGISTER WRITE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	$\overline{CS}$, A_0 Setup to $\overline{WR}$↓	0		ns	
t_{WA}	$\overline{CS}$, A_0 Hold After $\overline{WR}$↑	0		ns	
t_{WW}	$\overline{WR}$ Pulse Width	250		ns	
t_{DW}	Data Setup to $\overline{WR}$↑	150		ns	
t_{WD}	Data Hold After $\overline{WR}$↑	0		ns	

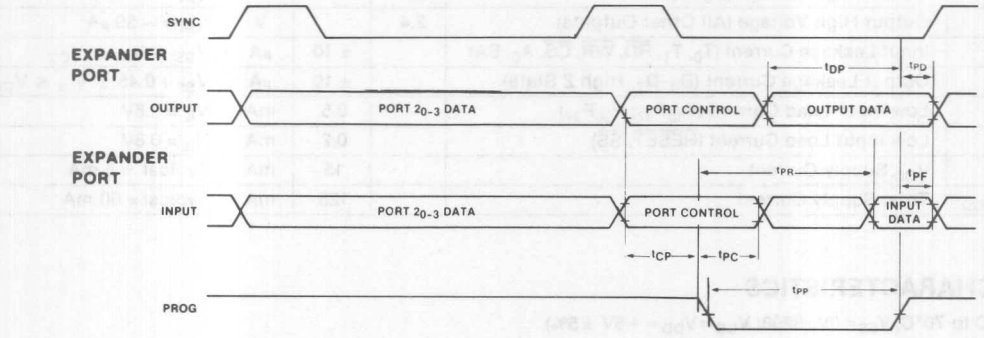
APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS

A.C. CHARACTERISTICS—PORT 2

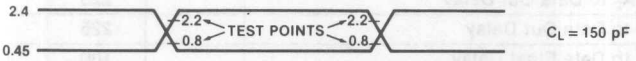
(T_A=0°C to 70°C: 8298; V_{CC}=+5V ±5%)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of PROG	110		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	100		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
t_{PF}	Input Data Hold Time	0	150	ns	
t_{DP}	Output Data Setup Time	250		ns	
t_{PD}	Output Data Hold Time	65		ns	
t_{PP}	PROG Pulse Width	1200		ns	

PORT 2 TIMING



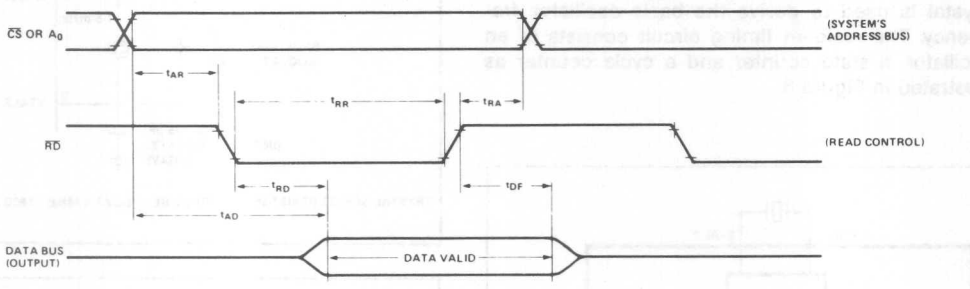
INPUT AND OUTPUT WAVEFORMS FOR A.C. TESTS



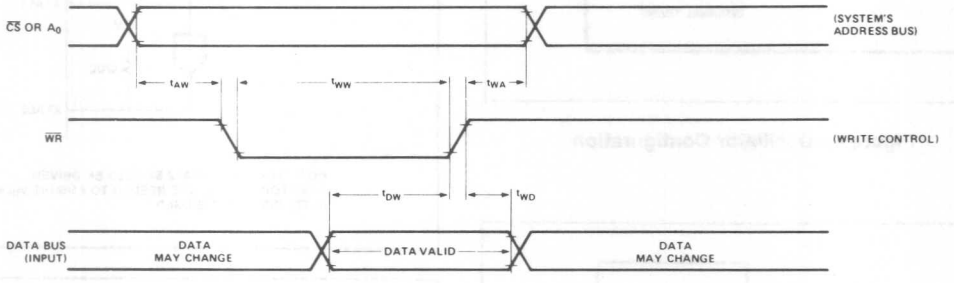
APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS

WAVEFORMS

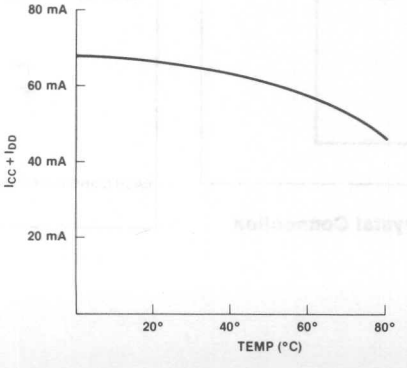
REGISTER READ OPERATION



REGISTER WRITE OPERATION



TYPICAL 8298 CURRENT



AFN-02027A

APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS

OSCILLATOR AND TIMING CIRCUITS

The 8298's internal timing generation is controlled by a self-contained oscillator and timing circuit. A 6 MHz crystal is used to derive the basic oscillator frequency. The resident timing circuit consists of an oscillator, a state counter and a cycle counter as illustrated in Figure 6.

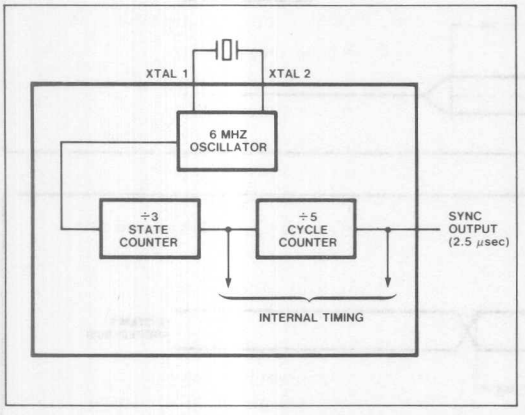


Figure 6. Oscillator Configuration

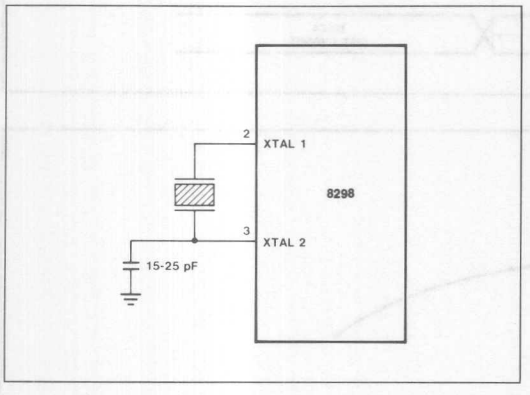
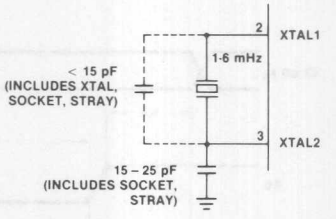


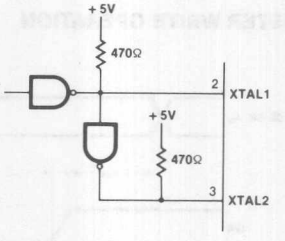
Figure 7. Recommended Crystal Connection

CRYSTAL OSCILLATOR MODE



CRYSTAL SERIES RESISTANCE SHOULD BE <75Ω AT 6 MHz; <180Ω AT 3.6 MHz.

DRIVING FROM EXTERNAL SOURCE

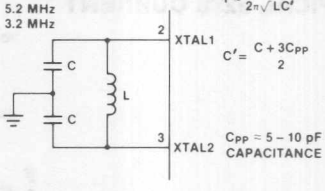


BOTH XTAL1 AND XTAL2 SHOULD BE DRIVEN. RESISTORS TO V_{CC} ARE NEEDED TO ENSURE $V_{IH} = 3.8V$ IF TTL CIRCUITRY IS USED.

LC OSCILLATOR MODE

L	C	NOMINAL f
45 μH	20 pF	5.2 MHz
120 μH	20 pF	3.2 MHz

$$f \approx \frac{1}{2\pi\sqrt{LC'}}$$



$$C' = \frac{C + 3C_{PP}}{2}$$

$C_{PP} \approx 5 - 10$ pF PIN-TO-PIN CAPACITANCE

EACH C SHOULD BE APPROXIMATELY 20 pF, INCLUDING STRAY CAPACITANCE

APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS

PROGRAMMING, VERIFYING, AND ERASING THE 8298 EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

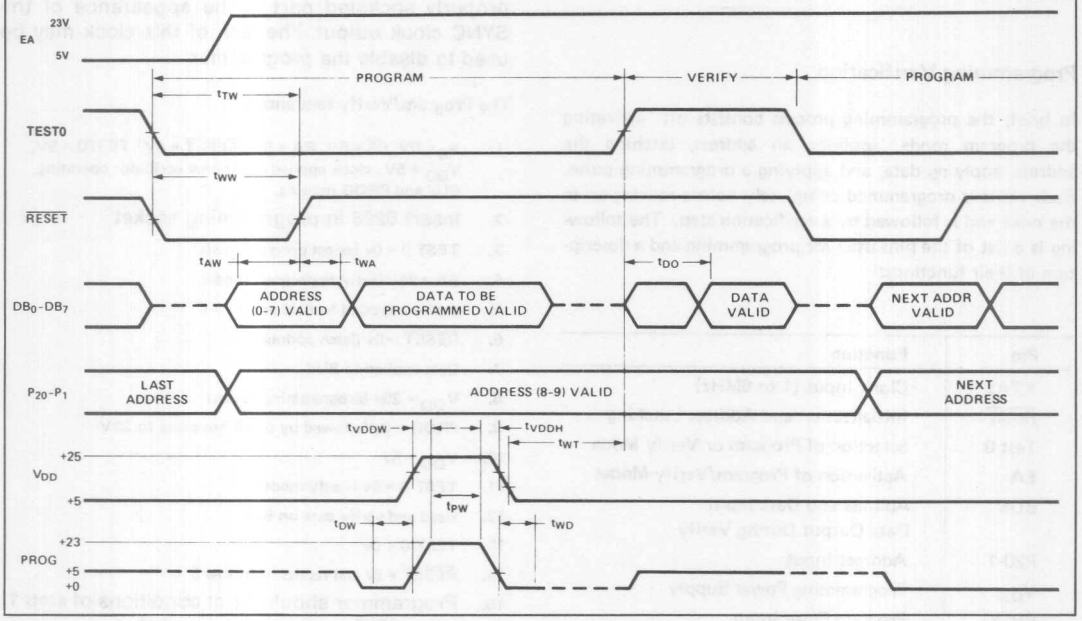
An attempt to program a missocketed 8298 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the SYNC clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

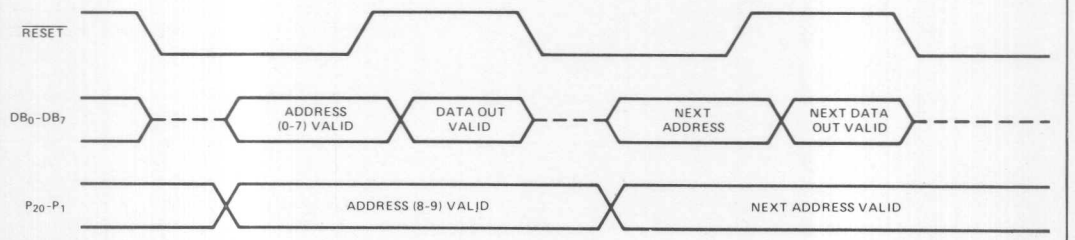
1. A_0 = 0V, $\overline{CS}$ = 5V, EA = 5V, $\overline{RESET}$ = 0V, TEST0 = 5V, V_{DD} = 5V, clock applied or internal oscillator operating, BUS and PROG floating.
2. Insert 8298 in programming socket
3. TEST 0 = 0v (select program mode)
4. EA = 23V (activate program mode)
5. Address applied to BUS and P20-1
6. $\overline{RESET}$ = 5v (latch address)
7. Data applied to BUS
8. V_{DD} = 25v (programming power)
9. PROG = 0v followed by one 50ms pulse to 23V
10. V_{DD} = 5v
11. TEST 0 = 5v (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0v
14. $\overline{RESET}$ = 0v and repeat from step 5
15. Programmer should be at conditions of step 1 when 8298 is removed from socket.

APPENDIX A: 8298 ELECTRICAL CHARACTERISTICS
WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. PROG MUST FLOAT IF EA IS LOW (i.e., =23V), OR IF T0 =5V FOR THE 8298.
2. XTAL1 AND XTAL2 DRIVEN BY 3.6 MHz CLOCK WILL GIVE 4.17 μsec t_{CY}. THIS IS ACCEPTABLE FOR 8298 PARTS AS WELL AS STANDARD PARTS.
3. AO MUST BE HELD LOW (i.e., =0V) DURING PROGRAM/VERIFY MODES.

The 8298 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP series) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

APPENDIX B: 8243 ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . . . 0°C to 70°C
Storage Temperature . . . . . -65°C to +150°C
Voltage on Any Pin
With Respect to Ground . . . . . -0.5V to +7V
Power Dissipation . . . . . 1 Watt

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

(8243: T_A = 0°C to 70°C, V_{CC} = 5V ± 10%)

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage Ports 4-7			0.45	V	I_{OL} = 5 mA*
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4-7	2.4			V	I_{OH} = 240μA
I_{IL1}	Input Leakage Ports 4-7	-10		20	μA	V_{in} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, CS, PROG	-10		10	μA	V_{in} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			45	V	I_{OL} = 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	
V_{OH2}	Output Voltage Port 2	2.4				I_{OH} = 100μA
I_{OL}	Sum of all IO_L from 16 Outputs			80	mA	5 mA Each Pin

*See following graph for additional sink current capability

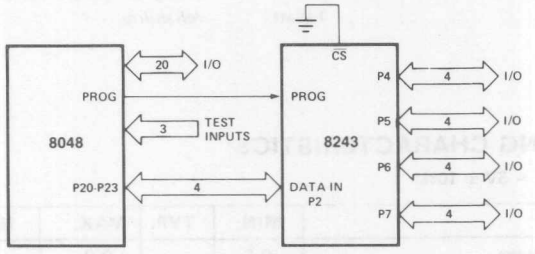
A.C. CHARACTERISTICS

(8243: T_A = 0°C to 70°C, V_{CC} = 5V ± 10%)

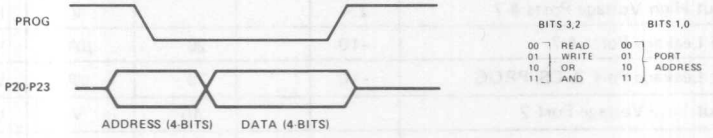
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_A	Code Valid Before PROG	100		ns	80 pF Load
t_B	Code Valid After PROG	60		ns	20 pF Load
t_C	Data Valid Before PROG	200		ns	80 pF Load
t_D	Data Valid After PROG	20		ns	20 pF Load
t_H	Floating After PROG	0	150	ns	20 pF Load
t_K	PROG Negative Pulse Width	700		ns	
t_{CS}	CS Valid Before/After PROG	50		ns	
t_{PO}	Ports 4-7 Valid After PROG		700	ns	100 pF Load
t_{LP1}	Ports 4-7 Valid Before/After PROG	100		ns	
t_{ACC}	Port 2 Valid After PROG		650	ns	80 pF Load

APPENDIX B: 8243 ELECTRICAL CHARACTERISTICS

EXPANDER INTERFACE

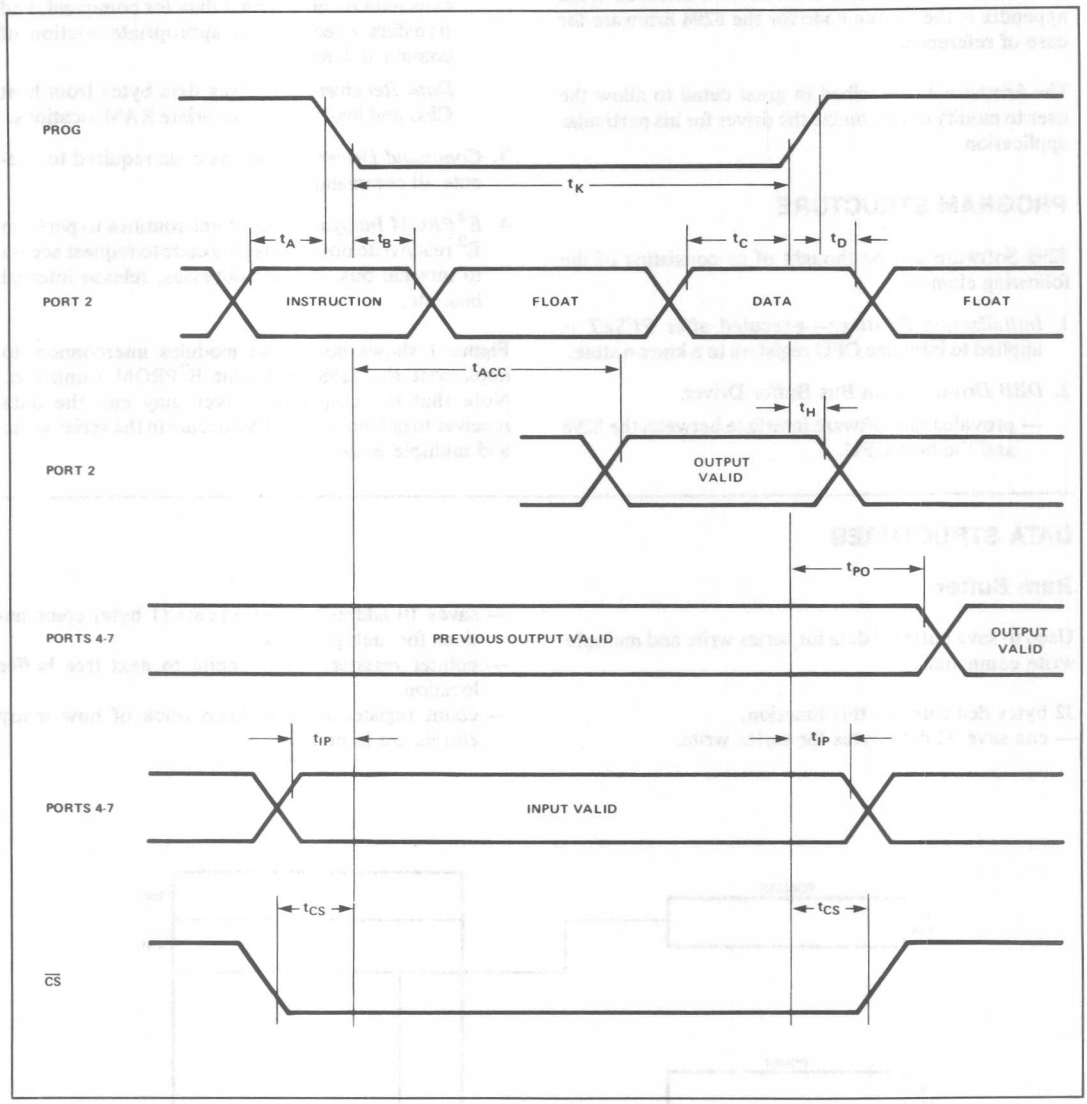


OUTPUT EXPANDER TIMING



APPENDIX B: 8243 ELECTRICAL CHARACTERISTICS

WAVEFORM



AFN-02027A

INTRODUCTION

The following is a description of the Firmware used in the 8298 E^2PROM Interface Controller. Included in the appendix is the source code for the 8298 firmware for ease of reference.

The firmware is described in great detail to allow the user to modify or customize the driver for his particular application.

PROGRAM STRUCTURE

8298 Software can be thought of as consisting of the following elements:

1. *Initialization Routines*—executed after *RESET* is applied to initialize CPU registers to a known state.
2. *DBB Driver*—Data Bus Buffer Driver.
   — provides the software interface between the 8298 and the host CPU.
   — handles receiving all commands and data.
   — DBB driver consists of two sections:

   Command Interpreter—Receives commands, calls data receiver to get data for command, and transfers execution to appropriate section of command driver.

   Data Receiver—Receives data bytes from host CPU and loads into appropriate RAM locations.
3. *Command Driver*—Contains code required to execute all commands.
4. *E^2PROM Interface*—Contains routines to perform E^2 read/write operations plus code to request access to internal bus, output addresses, release internal bus, etc.

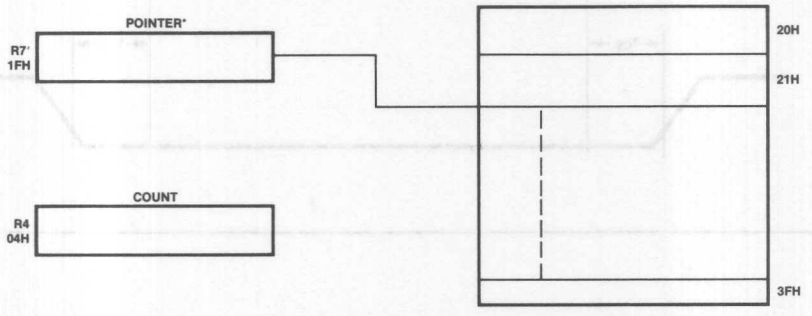
Figure 1 shows how these modules interconnect to implement the 8298 intelligent E^2PROM Controller. Note that the command driver may call the data receiver to get more data. This occurs in the series write and multiple write.

DATA STRUCTURES

Ram Buffer

Used to save buffered data for series write and multiple write commands.

32 bytes dedicated to this function.
— can save 32 data bytes for series write.
— saves 10 address (2 bytes)/ data (1 byte) combinations for multiple write.
— pointer register used to point to next free buffer location.
— count register used to keep track of how many entries are in buffer.



*POINTER IS ACTUALLY DATA DESTINATION REGISTER.

AFN-02027A

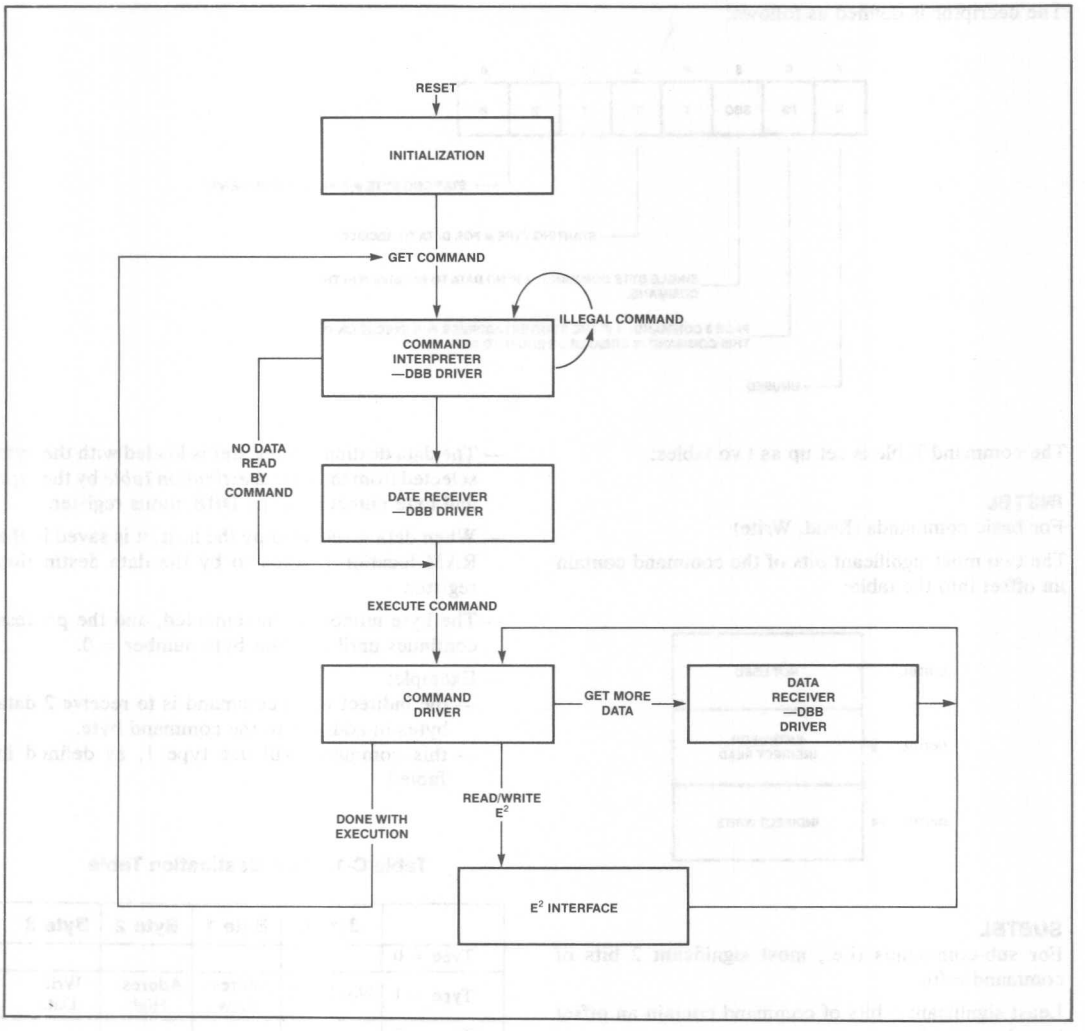
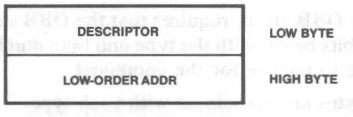


Figure C-1. 8298 Software Execution

Command Table

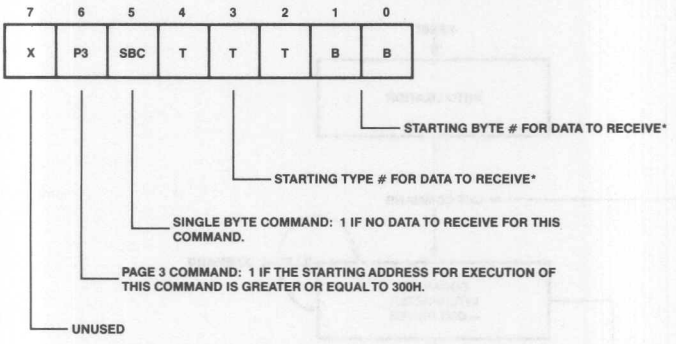
Contains entries for each command. Command table is located in ROM.

Each entry in command table consists of 2 bytes:



where the low order address is the low byte of the starting address for execution of the specified command.

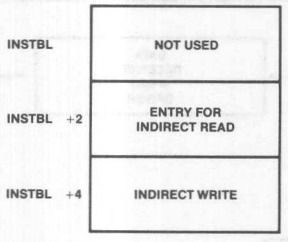
The decriptor is defined as follows:



The command Table is set up as two tables:

INSTBL
For basic commands (Read, Write).

The two most significant bits of the command contain an offset into the table:



SUBTBL
For sub-commands (i.e., most significant 2 bits of command = 0).

Least significant 6 bits of command contain an offset into the sub-table.

Data Destination Table

— Contains pointers to various RAM locations to be loaded with data received for command.

— The DBB driver requires that the DBB status register bits be set with the type and byte numbers of the data to receive for the command.

— 4 bytes are associated with each type.

— 7 bytes are available.

— A routine initializes the DBB status register with the starting byte number and the type of information to get.

— The data destination register is loaded with the byte selected from the *Data Destination Table* by the type and byte numbers in the DBB status register.

— When data is received by the host, it is saved in the RAM location pointed to by the data destination register.

— The byte number is incremented, and the process continues until the 2-bit byte number = 0.

— Example:
  — the indirect write command is to receive 2 data bytes in addition to the command byte.
  — this command will use type 1, as defined in Table 1.

Table C-1. Data Destination Table

	Byte 0	Byte 1	Byte 2	Byte 3
Type = 0				
Type = 1	Not Used	Address Low	Address High	Write Data
Type = 2				

— The descriptor for this command sets Type = 1, Byte = 2.

— The DBB driver will then load RAM with the data bytes received as follows:

1st byte	High address register
2nd byte	Data to Write

— In a similar vein, a routine may initialize the DBB status register with a type and byte number and call the GETDAT subroutine to get up to 4 bytes from the host and save in appropriate RAM locations.

Note: Type 7 is a special type which has the following attributes:

1. The data destination register is *not* loaded from the data destination table. The calling program initializes the data destination register
2. The data destination register is *incremented after* data is received.
3. 1–4 bytes are still received in this manner as selected by the starting byte number.

Figure 2 contains a graphical description of how registers are loaded.

BUS DESCRIPTION OF SOFTWARE ROUTINES

RESET

Entered upon external RESET signal.
a) zeroes: — DBB status register
— E^2 status register
b) initializes default write cycle time.
c) sets 'waiting for command' status.
d) continues w/GETCMD routine.

GETCMD

Called to get a command from user.
a) sets waiting for command bit in DBB status register.
b) optionally sets direct write possible bit
c) exits to GETDBB

CMDCPL

Command Complete—called at completion of command.
a) stops timer/counter in case running.
b) calls RELEAS to release internal bus.
c) enables OBF/IBF external host interrupts.
d) exits to GETCMD

ILLCMD

Illegal Command
a) set 'illegal command' status.
b) exits to GETCMD

DBBI

DBB interrupt service—actually, DBB interrupts are not used, but DBBI is called whenever a command or data is received from the host.
a) disables timer interrupt
b) sets STS = 0

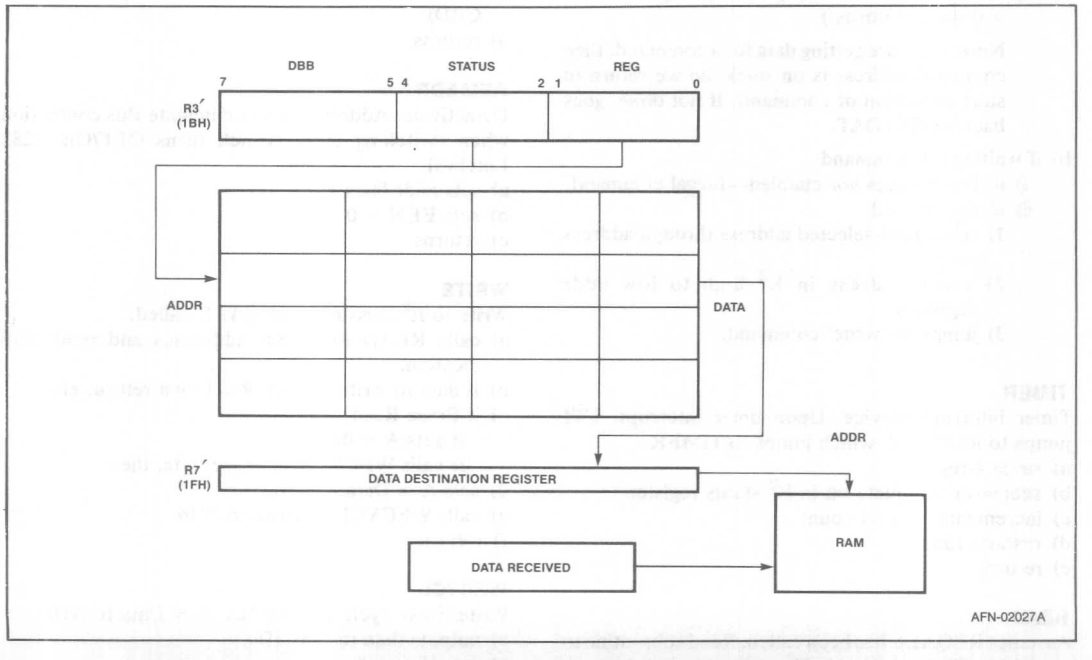


Figure C-2. Set-Up of Data Destination Information

c) determines if command or data received. If *command:*
   i) looks up command in command table, saves descriptor in DBB status register.
   ii) saves command byte in high address register if command is Read or Write.
   iii) saves address of command execution on stack.
   iv) jumps to command execution (via subroutine return) if one-byte command.
   v) continues w/GETDAT routine if *not* one-byte command.
d) if data received, goes to DATRCL

GETDAT
Get Data
a) Looks data destination from R3' (DBB status register) and data destination table. Places address in data destination register (R7).
   — exception: type = 111 —— no look up.
b) Enables external host interrupt.
c) Waits until data received and jumps to DBBI.

DATRCV
If data received from host:
a) if not waiting for command:
   i) saves data in location determined by data destination register (R7).
   ii) increments DBB status, if done, (Byte number = 0 then 'Returns')

   Note: If we are getting data for a command, then command address is on stack, so we return to start execution of command. If not done, goes back to GETDAT.

b) if waiting for command:
   i) if direct writes *not* enabled—illegal command.
   ii) if d/w enabled:
      1) reads host-selected address through address ports.
      2) saves address in E^2 high to low addr registers.
      3) jumps to 'write' command.

TIMER
Timer interrupt service. Upon timer interrupt, UPI jumps to location 7, which jumps to TIMER.
a) saves 4-reg.
b) sets write complete bit in E^2 status register.
c) increments internal count
d) restarts timer
e) returns

READ
Assumes REQALL has been called. Read subroutine to read from 2816. Address in E^2 addr registers. Returns data read in A-Reg.
a) outputs Address (Call OUTADR)
b) set UPI-RD I/O line = 0
c) read data through P1
d) sets UPI-RD I/O line = 1
e) returns

OUTADR
Output addresses from E^2 registers to I/O lines. (1A 0–1A 14)
a) gets high order address.
b) i) shifts MSBs left by 2 so they are in Bits 7, 6
   ii) outputs to Port 2, so IA14 = 27, IA13 = P26.
c) outputs low order address bits
d) outputs bits IA8–IA11.
e) leaves EEN = 1.

REQALL
Request access to local bus and waits until access acknowledged.
a) if access received (T0 = 1) then returns, else:
b) sets EEREQ = 1, EEREQ = 0
c) waits until T0 = 1, (EEACK = 1) and returns,

RELEAS
Release control of local bus
a) sets all Address lines = 1
b) sets all data lines = 1
c) de-activates all control lines (EEREQ, EEN, VPP, CRD)
d) returns

REMADR
De-activates Address lines to eliminate Bus contention when switching EEN (which turns OFF/On, 8282 Latches)
a) sets addr lines = 1
b) sets EEN = 0
c) returns

WRITE
Write to E^2—assumes REQALL called.
a) calls READ to output addresses and read 2816 location.
b) if data to write = Data Read then return, else
c) if Erase Read, then:
   i) sets A = 0FFH
   ii) calls WECYCL to Erase byte, then
d) sets A = Data to Write
e) calls WECYCL to write to 2816
f) returns

WECYCL
Write/Erase cycle subroutines. A = Data to Write.
a) outputs data to ID0–ID7.
b) sets V_{PP} = 0
c) resets write complete bit of E^2 status register.

APPENDIX C: 8298 FIRMWARE DESCRIPTION

d) starts timer
e) waits until write complete bit set by timer interrupt service routine. Calls CKDBB to check for abort command while waiting.
f) at end of write cycle, calls SHUT to shut down V_{PP} switch
g) returns

SHUT
Turns off V_{PP} switch
a) set V_{PP} = 0, UPP = 1
b) waits a time of 100 μs for V_{PP} signal to fall to 5V
c) returns

CKDBB
Check data Bus Buffer for abort command.
a) If buffer not full THEN returns ELSE
If not command in buffer (F1 = 0)
THEN returns
ELSE If abort command THEN GOTO abort routine
ELSE — set illegal command status
— return

INCADR
Increments E^2 high, low address register pair.

DECCNT
Decrements count register pair. If zero, returns with REG = 0

Command Driver

The following are descriptions of routines which execute commands. When these routines are called, data for the selected command has already been received.

READC
Read Command
a) Requests access to local bus (REQACC)
b) Calls read Subroutine
c) Calls OUTPUT to output result to host

OUTPUT
Called to place data in A-Reg into output buffer. Waits until host has read output buffer before exiting.

WRITEC
Write Command
a) calls REQACC to request access to local bus.
b) calls write subroutine

CEREASE
Chip Erase Command
a) requests access to bus
b) moves high order address bits, IAB, IA14 to P26, P27.
c) outputs high order address to P1 with bit 7 = 0 (CERASE) = 0
d) pulses EEN to V_{OH} to latch high address and chip erase signal.
e) calls WECYCL with A-REG = 0FFH to erase a chip
f) calls REMCE to remove chip erase signal

REMCE
Remove chip erase
a) sets P1 = 0FFH
b) sets EEN = 0
c) sets EEN = 1 to latch. No chip erase
d) exits

BLOCKE
Block erase command
a) requests access
b) sets data to write = 0FFH (erase)
c) writes to E^2
d) remove address to prevent bus contention
e) increment address
f) decrements count, if not done go to step b)

MULTWR
Multiple write command
a) zeroes buffer count and initializes buffer pointer
b) calls GETDAT to get 10 address/data combinations or # combinations left to get, whichever is lower.
— Data is loaded into RAM buffer
c) dumps buffer to E^2.
Note: Data is saved in RAM as follows:
High Addr
Low Addr
Data to Write
d) continues to step b) if not done

SERWRT
Series write command
a) zeroes buffer counter and initializes buffer pointer.
b) gets up to 32 bytes of data or # bytes left to get, whichever is less.
c) requests access, dumps data in buffer to E^2 memory. Addresses are incremented after each byte is written.
d) continues to step b) if not done

SERRD
Series read command
a) requests access to bus
b) reads a byte
c) outputs it to host
d) increments address
e) decrements count
f) continues to step b) if not done

APPENDIX C: 8298 FIRMWARE DESCRIPTION

ABORT

a) If V_{PP} is on (*UPP* = 0)
   i) calls SHUT to turn off V_{PP}.
b) calls REMCE to remove chip erase in case activated.

READAL

Read last low address.

READAH

Read last high address.

RADWR

Read last data to write to E^2.

APPENDIX D: 8298 E^2PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT
                   1 $MOD41A MACROFILE DEBUG
                   2
                   3
                   4
                   5         ;
                   6         ;
                   7         ;
                   8         ;
                   9         ;
                  10         ;
                  11         ;
                  12         ;
                  13         ;
                  14         ;
                  15         ;
                  16
                  17
                  18
                  19
                  20         ;       UPI REGISTER DEFINITIONS
                  21
                  22         ;       RB0:
                  23
0000              24 SCR0     EQU     0       ; R0 = SCRATCH
0001              25 SCR1     EQU     1       ; R1 = SCRATCH
0002              26 SCR2     EQU     2       ; R2 = SCRATCH
0003              27 INTCNT   EQU     3       ; R3 = INTERVAL COUNT
0004              28 BUFCNT   EQU     4       ; R4 = BUFFER COUNTER
0005              29 WRTDAT   EQU     5       ; R5 = DATA TO WRITE
0006              30 CNTLO    EQU     6       ; R6 = LOW ORDER COUNT
0007              31 CNTHI    EQU     7       ; R7 = HIGH ORDER COUNT
                  32
                  33
                  34         ;       RB1:
                  35
0018              36 SCR0P    EQU     18H     ; R0' = SCRATCH
0019              37 SCR1P    EQU     19H     ; R1' = SCRATCH
001A              38 COMFB    EQU     1AH     ; R2' = COMMAND FIRST BYTE
001B              39 DBSTAT   EQU     1BH     ; R3' = DBB STATUS REGISTER
001C              40 ASAVE    EQU     1CH     ; R4' = A-REGISTER SAVE
001D              41 EESTAT   EQU     1DH     ; R5' = EE STATUS REGISTER
001E              42 INTIME   EQU     1EH     ; R6' = INITIAL WIRTE TIMER COUNT
001F              43 DATDES   EQU     1FH     ; R7' = DATA DESTINATION REGISTER
                  44
                  45 $EJECT
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT
                  46
                  47        ;       NON-REGISTER MEMORY DEFINITIONS
                  48
0014              49 ADRLO  EQU     14H     ; LOW ORDER EE ADDRESS
0015              50 ADRHI  EQU     15H     ; HI ORDER EE ADDRESS
0020              51 BUFST  EQU     20H     ; BUFFER START ADDRESS
0016              52 ININT  EQU     16H     ; INITIAL INTERVAL COUNT
003E              53 MWBEND EQU     20H+30D ; MULTIPLE WRITE BUFFER END ADDRESS
0040              54 SWBEND EQU     20H+32D ; SERIAL WRITE BUFFER END ADDRESS
                  55
                  56
                  57        ;       I/O DEFINITIONS
                  58
                  59        ;       HOST INTERRUPTS
                  60
00DF              61 IBFINA EQU     11011111B       ; INACTIVE IBF' HOST INTERRUPT
FF20              62 IBFACT EQU     NOT IBFINA      ; ACTIVE IBF' HOST INTERRUPT
0010              63 OBFACT EQU     00010000B       ; ACTIVE OBF HOST INTERRUPT
FFEF              64 OBFINA EQU     NOT OBFACT      ; INACTIVE OBF HOST INTERRUPT
                  65
                  66        ;       CONTROL LINES
                  67
000E              68 VPPACT EQU     1110B           ; ACTIVE VPP SWITCH
FFF1              69 VPPINA EQU     NOT VPPACT      ; INACTIVE VPP SWITCH
0007              70 REQACT EQU     0111B           ; EEREQ ACTIVE SIGNAL
0009              71 RDACT  EQU     1001B           ; ACTIVATE UPIRD' AND EEN'
0002              72 RDINA  EQU     0010B           ; DE-ACTIVATE UPIRD'
000B              73 EENACT EQU     1011B           ; ACTIVATE EEN'
0004              74 EENINA EQU     0100B           ; DEACTIVATE EEN'
                  75
                  76        ;       STATUS DEFINITIONS
                  77
0007              78 WIPSTS EQU     0111B           ; WRITE-IN-PROGRESS STATUS
                  79
                  80
                  81        ;       TIME VALUES
                  82
FF83              83 TIME   EQU     -125D           ; TIMER COUNT VALUE FOR 3.333 MSEC TIME COUNT
007F              84 INITWR EQU     7FH             ; INIT EESTAT WRT VALUE FOR WRT COMPL BIT =0
000E              85 VPPFALL EQU     14D            ; VPP FALL TIME.  SET FOR > 100 USEC
                  86                                ; FOR WRITE CYCLE TIMER COUNT
                  87
                  88
                  89        ;       MISC
                  90
000A              91 ABORTC EQU     0AH             ; ABORT COMMAND CODE
                  92
                  93 $EJECT
```

APPENDIX D: 8298 E^2PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT
                   94
                   95 ;              REGISTER FORMATS:
                   96
                   97
                   98 ;        1) DBB STATUS REGISTER (R3' = 1BH)
                   99 ;
                  100 ;                 BIT(S)          DESCRIPTION
                  101 ;                 ------          -----------
                  102 ;                 0-1             NEXT BYTE # TO GET (0-3)
                  103 ;                 2-4             CURRENT TYPE OF BYTE (FROM DATA DEST TABLE)
                  104 ;                  5              WAITING FOR COMMAND (=1)
                  105 ;                  6              DIRECT WRITE ENABLED (=1)
                  106 ;
                  107 ;
                  108 ;        2) EE STATUS REGISTER (R5' = 1DH)
                  109 ;
                  110 ;                 BIT(S)          DESCRIPTION
                  111 ;                 ------          -----------
                  112 ;                 0-3             INTERVAL LOW ORDER COUNT (0-3)
                  113 ;                  7              WRITE COMPLETE INTERRUPT RECEIVED
                  114 ;
                  115 ;
                  116 ;        3) F0 = DIRECT WRITE POSSIBLE
                  117 ;
                  118 ;        4) FLAG F1 = COMMAND RECEIVED (=1)
                  119 ;
                  120
                  121 $EJECT
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
                  122
                  123         ;       INITIALIZATION
                  124
0000              125         ORG     0
                  126
0000 4609         127 RESET:  JNT1    INIT
                  128
0007              129         ORG     7H
0007 04FC         130         JMP     TIMER           ; TIMER INTERRUPT VECTOR
                  131
                  132 INIT:
0009 F5           133         EN      FLAGS           ; ENABLE HOST INTERRUPT FLAGS
000A 85           134         CLR     F0              ; SET F0 = 0 TO INDICATE DIRECT WRITE NOT POSS
000B D5           135         SEL     RB1             ; SELECT ALTERNATE REG SET TEMPORARILY
000C 27           136         CLR     A               ; SET A = 0
000D AB           137         MOV     R3,A            ; ZERO DBB STATUS REGISTER
000E AD           138         MOV     R5,A            ; ZERO EE STATUS REGISTER
000F BE83         139         MOV     R6,#TIME        ; INITIAL WRITE TIMER VALUE
0011 C5           140         SEL     RB0             ; SELECT NORMAL REGISTER SET
0012 23F0         141         MOV     A,#0F0H         ; OUTPUT WAITING FOR COMMAND STATUS
0014 90           142         MOV     STS,A
                  143
                  144 GETCMD:
0015 A5           145         CLR     F1              ; RESET LAST STATE OF COMMAND/DATA REGISTER
0016 D5           146         SEL     RB1             ; SELECT ALTERNATE REG BANK
0017 FB           147         MOV     A,R3            ; GET DBB STATUS
0018 4320         148         ORL     A,#20H          ; OR IN WAITING FOR COMMAND BIT
001A 85           149         CLR     F0              ; CLEAR DIRECT WRITE POSSIBLE BIT
001B D21F         150         JB6     DWE             ; ONLY SET D/W POSSIBLE IF D/W ENABLED
001D 0420         151         JMP     DWNE            ; D/W NOT ENABLED -DON'T SET BIT
                  152 DWE:
001F 95           153         CPL     F0              ; SET D/W POSSIBLE BIT
                  154 DWNE:
0020 AB           155         MOV     R3,A            ; RE-SAVE DBB STATUS REG
0021 04B3         156         JMP     GETDBB          ; GO GET SOME MORE DATA
                  157
                  158
                  159         ;       COMMAND COMPLETE
                  160
                  161 CMDCPL:
0023 C5           162         SEL     RB0             ; SELECT NORMAL REG SET
0024 65           163         STOP    TCNT            ; STOP ANY TIMER ACTIVITY
0025 3430         164         CALL    RELEAS          ; RELEASE CONTROL OF BUS
0027 23F0         165         MOV     A,#0F0H         ; SET COMMAND COMPLETE STATUS
0029 90           166         MOV     STS,A
002A 8A30         167         ORL     P2,#00110000B   ; ALLOW OBF,IBF HOST INTERRUPTS
002C 0415         168         JMP     GETCMD          ; GO GET ANOTHER COMMAND
                  169
                  170         ;       ILLEGAL COMMAND:
                  171
                  172 ILLCMD:
002E 23E0         173         MOV     A,#11100000B    ; SET ILLEGAL COMMAND STATUS
0030 90           174         MOV     STS,A
0031 0415         175         JMP     GETCMD
                  176
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT
                 178
                 179        ;       DBB INTERRUPT SERVICE ROUTINE AND DATA RETRIEVER
                 180
                 181 DBBI:
0033 35          182        DIS     TCNTI           ; DISABLE INTERRUPTS
0034 27          183        CLR     A               ; ZERO STS REG BECAUSE WE'RE NO LONGER LOOKING
                 184                                ; FOR COMMAND/DATA
0035 90          185        MOV     STS,A
0036 D5          186        SEL     RB1             ; SELECT THIS REGISTER SET
0037 AC          187        MOV     R4,A            ; SAVE A-REGISTER
0038 9ADF        188        ANL     P2,#IBFINA      ; DE-ACTIVATE IBF INTERRUPTS TO HOST
003A 85          189        CLR     F0              ; CLEAR DWP BIT
003B 763F        190        JF1     DBBI07          ; IF COMMAND RECEIVED, CONTINUE, ELSE
003D 04C7        191        JMP     DATRCV          ; COMMAND - GO PROCESS THE COMMAND
                 192 DBBI07:
003F 22          193        IN      A,DBB           ; GET COMMAND
0040 AA          194        MOV     R2,A            ; SAVE IT
0041 53F0        195        ANL     A,#0F0H         ; CHECK IF ILLEGAL COMMAND (30H - 3FH)
0043 D24D        196        JB6     NOTICD          ; TO SAVE SPACE, PART OF TABLE ELIMINATED
0045 F24D        197        JB7     NOTICD          ; MUST HAVE B5 AND B4 = 1 WITH B6 AND B7
0047 924B        198        JB4     MAYBIC          ; = 0 TO HAVE AN ILLEGAL COMMAND.
0049 044D        199        JMP     NOTICD
                 200 MAYBIC:
004B B22E        201        JB5     ILLCMD          ; YES, IT IS ILLEGAL. GO DEAL WITH IT
                 202 NOTICD:
004D 53C0        203        ANL     A,#0C0H         ; MASK ALL BUT HI ORDER BITS
004F C675        204        JZ      SUBCMD          ; IF HI ORDER BITS ZERO THEN SUB COMMAND
0051 E7          205        RL      A               ; MOVE HI BITS TO LOW BITS
0052 E7          206        RL      A               ; AND MULTIPLY BY 2 IN THE PROCESS
0053 E7          207        RL      A
0054 0365        208        ADD     A,#LOW INSTBL   ; LET A POINT TO ENTRY IN INSTRUCTION TABLE
                 209 LOOKUP:
0056 A8          210        MOV     R0,A            ; SAVE POINTER TEMPORARILY
0057 E3          211        MOVP3   A,@A            ; GET COMMAND DESCRIPTOR BYTE FROM TABLE
0058 28          212        XCH     A,R0            ; SWAP DESCRIPTOR FOR TABLE ADDRESS
0059 17          213        INC     A               ; POINT TO COMMAND START ADDRESS IN TABLE
005A E3          214        MOVP3   A,@A            ; GET COMMAND START ADDRESS
005B 28          215        XCH     A,R0            ; SWAP DEST ADDR FOR COMMAND DESCRIPTOR
005C A9          216        MOV     R1,A            ; SAVE DESCRIPTOR IN R1
005D 531F        217        ANL     A,#1FH          ; MASK TO KEEP DATA TO RECEIVE
005F 2B          218        XCH     A,R3            ; SWAP A & DBB STATUS
0060 53E0        219        ANL     A,#0E0H         ; REMOVE LOW ORDER BITS
0062 4B          220        ORL     A,R3            ; ADD IN DATA DESCRIPTOR
0063 AB          221        MOV     R3,A            ; SAVE UPDATED DBB REG
0064 B27D        222        JB5     PCMND           ; IF WAITING FOR COMMAND BIT SET GO PROCESS IT
0066 23F0        223        MOV     A,#-1*(LOW ABORT); WE'RE NOT WAITING FOR COMMAND, SO THE ONLY
                 224                                ; COMMAND WE'LL ACCEPT IS ABORT - CHECK FOR
                 225                                ; ABORT COMMAND
0068 68          226        ADD     A,R0
0069 966D        227        JNZ     ILLCDO          ; NOT ABORT - GO INDICATE ILLEGAL COMMAND
006B 6410        228        JMP     ABORT           ; YES ABORT - GO PROCESS IT
                 229
                 230        ;       ILLEGAL COMMAND DURING WRITE CYCLE
                 231
                 232 ILLCDO:
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ       LINE      SOURCE STATEMENT
006D 23AA       233       MOV     A,#0AAH         ; OUTPUT ILLEGAL COMMAND MESSAGE
006F 02         234       OUT     DBB,A
0070 8910       235       ORL     P1,#0BFACT      ; ALLOW OBF EXTERNAL INTERUPTS
                236 EXIT:
0072 FC         237       MOV     A,R4            ; RESTORE A-REGISTER
0073 25         238       EN      TCNTI           ; RE-ENABLE TIMER INTERRUPT
0074 93         239       RETR                    ; RETURN TO USER PROGRAM
                240
                241       ;       LOOK UP SUB COMMAND AND POINT TO ENTRY IN TABLE
                242
                243 SUBCMD:
0075 FA         244       MOV     A,R2            ; GET COMMAND
0076 533F       245       ANL     A,#3FH          ; KEEP LOWER 6 BITS
0078 E7         246       RL      A               ; MULTIPLY BY 2 BECAUSE 2 BYTES PER INSTR
0079 036D       247       ADD     A,#LOW SUBTBL   ; ADD START ADDR OF SUB COMMAND TABLE
007B 0456       248       JMP     LOOKUP          ; GO LOOKUP SUB COMMAND
                249
                250
                251       ;       PROCESS THE COMMAND
                252
                253 PCMND:
007D A5         254       CLR     F1              ; CLEAR COMMAND FLAG
007E 27         255       CLR     A               ; ZERO STS BITS
007F 90         256       MOV     STS,A
                257 PC03:
0080 FB         258       MOV     A,R3            ; GET DBB STATUS
0081 53DF       259       ANL     A,#11011111B    ; CLEAR WAITING FOR COMMAND BIT
0083 AB         260       MOV     R3,A            ; RE-SAVE DBB STATUS
0084 C7         261       MOV     A,PSW           ; GET PSW
0085 2311       262       MOV     A,#00010001B    ; SET STACK POINTER = 1ST LOC
0087 D7         263       MOV     PSW,A           ; AND RESTORE PSW
0088 FA         264       MOV     A,R2            ; GET COMMAND FIRST BYTE IN A
0089 F2B9       265       JB7     SAVEHI          ; IF BIT 7 OR BIT 6 SET SAVE HI ADDRESS
008B D2B9       266       JB6     SAVEHI
                267 PC05:
008D F8         268       MOV     A,R0            ; SET COMMAND DEST ADDRESS IN STACK
008E B808       269       MOV     R0,#8H          ; IN 1ST LOCATION IN STACK
0090 A0         270       MOV     @R0,A
0091 F9         271       MOV     A,R1            ; GET COMMAND DESCRIPTOR
0092 D298       272       JB6     P3INST          ; IF PAGE 3 BIT SET ADD PAGE 3 ADDRESS
0094 2302       273       MOV     A,#HIGH FIRSTIN ; SET HI STACK BYTE FOR RB0 SELECTED
                274                               ; AND HIGH ORDER ADDRESS OF 1ST INSTRUCTION
0096 049A       275       JMP     PC10
                276 P3INST:
0098 2303       277       MOV     A,#HIGH FIRSTP3 ; SET HI STACK BYTE FOR RB0 SELECTED
                278                               ; AND HI ORDER ADDRESS OF 1ST PAGE 3 INSTR
                279 PC10:
009A 18         280       INC     R0              ; POINT TO HIGH ORDER STACK LOCATION
009B A0         281       MOV     @R0,A           ; SAVE ON STACK
009C F9         282       MOV     A,R1            ; GET COMMAND DESCRIPTOR
009D B2C3       283       JB5     EXEC            ; GO EXECUTE COMMAND IF NO DATA TO RECEIVE
                284
                285       ;       GET DATA (USER PROGRAM MAY ENTER HERE)
                286
                287 GETDAT:
```

APPENDIX D: 8298 E^2PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT
009F B900        288        MOV     R1,#0           ; ZERO STS DATA COUNTER
                 289 SETGET:
00A1 F9          290        MOV     A,R1            ; GET STS DATA COUNT
00A2 4308        291        ORL     A,#8H           ; OR-IN WAITING FOR COMMAND/DATA BIT
00A4 47          292        SWAP    A               ; MOVE TO HIGH ORDER BITS
00A5 90          293        MOV     STS,A           ; AND OUTPUT TO STS REGISTER
00A6 FB          294        MOV     A,R3            ; GET DBB STATUS
00A7 43E3        295        ORL     A,#11100011B    ; CHECK FOR TYPE = 111
00A9 37          296        CPL     A               ; IF TYPE = 111 THEN A-REG SHOULD BE 0
00AA C6B3        297        JZ      GETDBB          ; IF SPECIAL TYPE (=111) DON'T LOOK UP DEST
00AC FB          298        MOV     A,R3            ; GET DBB STATUS
00AD 531F        299        ANL     A,#1FH          ; KEEP ONLY TYPE AND BYTE BITS
00AF 03CD        300        ADD     A,#LOW DESTBL   ; ADD START OF DESTINATION TABLE
00B1 E3          301        MOVP3   A,@A            ; GET DESTINATION
00B2 AF          302        MOV     R7,A            ; AND SAVE IN DATA DESTINATION REGISTER
                 303 GETDBB:
00B3 8A20        304        ORL     P2,#IBFACT      ; ENABLE HOST INTERRUPT
00B5 D6B5        305 WAITB:  JNIBF   WAITB           ; WAIT UNTIL DATA RECEIVED
00B7 0433        306        JMP     DBBI            ; GO PROCESS DATA WE JUST GOT
                 307
                 308        ;       SAVE HI-ORDER ADDRESS FOR READ OR WRITE
                 309
                 310 SAVEHI:
00B9 29          311        XCH     A,R1            ; SAVE R1 IN R7
00BA 2F          312        XCH     A,R7
00BB B915        313        MOV     R1,#ADRHI       ; POINT TO HI ORDER EE ADDRESS
00BD FA          314        MOV     A,R2            ; GET COMMAND FIRST BYTE TO PUT IN HI ADDRESS
00BE A1          315        MOV     @R1,A           ; SAVE COMMAND IN HI ORDER ADDRESS REGISTER
00BF 2F          316        XCH     A,R7            ; RESTORE R1
00C0 A9          317        MOV     R1,A
00C1 048D        318        JMP     PC05            ; CONTINUE PROCESSING COMMAND
                 319
                 320        ;       EXECUTE COMMAND
                 321
                 322 EXEC:
00C3 27          323        CLR     A               ; CLEAR STS TO INDICATE PROCESSING
00C4 90          324        MOV     STS,A
00C5 0472        325        JMP     EXIT            ; AND EXIT.
                 326
                 327        ;       DATA RECEIVED
                 328
                 329 DATRCV:
00C7 22          330        IN      A,DBB           ; GET DBB DATA
00C8 A8          331        MOV     R0,A            ; SAVE TEMPORARILY IN R0
00C9 FB          332        MOV     A,R3            ; GET DBB STATUS
00CA B2DD        333        JB5     DWCMD           ; IF WE'RE WAITING FOR COMMAND THEN MAYBE D/W
00CC 2F          334        XCH     A,R7            ; SAVE A IN R7 AND GET DATA DEST PTR
00CD 28          335        XCH     A,R0            ; GET DATA TO SAVE IN A & ADDR IN R0
00CE A0          336        MOV     @R0,A           ; SAVE IN APPROPRIATE LOCATION
00CF 28          337        XCH     A,R0            ; RETRIEVE ADDRESS IN A
00D0 17          338        INC     A               ; INCREMENT DESTINATION ADDRESS IN CASE OF SPEC
                 339                                ; TYPE
00D1 2F          340        XCH     A,R7            ; SAVE IN R7 & RETRIEVE DBB IN A
00D2 17          341        INC     A               ; INCREMENT BYTE # TO GET
00D3 53DF        342        ANL     A,#11011111B    ; MAKE SURE WAITING FOR COMMAND BIT = 0
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
00D5 AB          343          MOV     R3,A            ; RE-SAVE DBB STATUS
00D6 5303        344          ANL     A,#3H           ; MASK OUT ALL BUT BYTE COUNT
00D8 C6C3        345          JZ      EXEC            ; IF ZERO THE WE'RE DONE - CONTINUE CMMND EXEC
00DA 19          346          INC     R1              ; INCREMENT DATA BYTE COUNTER
00DB 04A1        347          JMP     SETGET          ; NO - GO GET MORE DATA
                 348
                 349          ;       DIRECT WRITE COMMAND
                 350
                 351 DWCMD:
00DD D2E1        352          JB6     DW05            ; IF DIRECT WRITES ENABLED THEN GO PROCESS
00DF 042E        353          JMP     ILLCMD          ; ELSE ILLEGAL COMMAND RECEIVED
                 354
                 355 DW05:
00E1 2305        356          MOV     A,#WRTDAT       ; POINT TO WRITE DATA REGISTER
00E3 28          357          XCH     A,R0            ; SWAP
00E4 A0          358          MOV     @R0,A           ; SAVE DATA TO WRITE IN REGISTER
00E5 B814        359          MOV     R0,#ADRLO       ; POINT TO LOW ORDER ADDRESS REGISTER
00E7 8D          360          MOVD    A,P5            ; GET LOW ORDER ADDRESS FROM ADDRESS PORTS
00E8 47          361          SWAP    A
00E9 A9          362          MOV     R1,A            ; SAVE TEMPORARILY IN R1
00EA 8C          363          MOVD    A,P4
00EB 49          364          ORL     A,R1            ; BRING IN HI ORDER NIBBLE
00EC A0          365          MOV     @R0,A           ; SAVE IN LOW ORDER ADDRESS REGISTER
00ED 8E          366          MOVD    A,P6            ; GET HIGH ORDER ADDRESS
00EE 18          367          INC     R0              ; POINT TO HI ORDER ADDRESS
00EF A0          368          MOV     @R0,A           ; SAVE HI ORDER ADDRESS
00F0 8A          369          IN      A,P2            ; GET HI ORDER ADDRESS BITS
00F1 77          370          RR      A               ; MOVE TWO BITS TO THE RIGHT
00F2 77          371          RR      A
00F3 5330        372          ANL     A,#30H          ; MAKE SURE THAT'S ALL THE DATA WE HAVE
00F5 40          373          ORL     A,@R0           ; READ IN REST OF HI ORDER ADDRESS BYTE
00F6 A0          374          MOV     @R0,A           ; AND SAVE IN HI ORDER ADDRESS REGISTER
                 375 DW10:
00F7 2300        376          MOV     A,#00000000B    ; SET STACK PROINTER TO POINT TO
                 377                                  ; 2ND LEVEL STACK AND RB0 SELECTED
00F9 D7          378          MOV     PSW,A
00FA 4410        379          JMP     WRITEC          ; GO EXECUTE WRITE COMMAND
                 380
                 381 $EJECT
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT
                 382
                 383        ;       TIMER INTERRUPT
                 384
                 385 TIMER:
00FC D5          386        SEL     RB1             ; SELECT ALTERNATE REGISTER BANK
00FD AC          387        MOV     R4,A            ; SAVE A-REG
00FE 1D          388        INC     R5              ; INCREMENT INTERVAL COUNT
00FF FD          389        MOV     A,R5            ; GET EE STATUS
0100 4380        390        ORL     A,#80H          ; SET WRITE COMPLETE INTERRUPT BIT
0102 AD          391        MOV     R5,A            ; RESTORE EE STATUS
0103 FE          392        MOV     A,R6            ; RESET TIMER COUNT
0104 62          393        MOV     T,A
0105 FC          394        MOV     A,R4            ; RESTORE A-REG
0106 93          395        RETR                    ; BACK TO INTERUPTED PROGRAM
                 396
                 397
                 398
                 399        ;       READ / WRITE SUBROUTINES
                 400
                 401
                 402
                 403        ;       ************************************************
                 404
                 405        ;               EE READ - CALLED W/ ADDRESS IN ADDRESS REG
                 406
                 407        ;               DATA RETURNED IN A-REG
                 408        ;               USES:   R0
                 409        ;               ENABLES LOCAL BUS
                 410
                 411        ;               ASSUMES ACCESS TO LOCAL BUS
                 412
                 413        ;       ************************************************
                 414
                 415 READ:
0107 3413        416        CALL    OUTADR          ; OUTPUT EE ADDRESS
0109 2309        417        MOV     A,#RDACT        ; ACTIVATE READ LINE
010B 9F          418        ANLD    P7,A
010C 09          419        IN      A,P1            ; READ THE DATA
010D A8          420        MOV     R0,A            ; SAVE TEMPORARILY
010E 2302        421        MOV     A,#RDINA        ; DE-ACTIVATE READ LINE
0110 8F          422        ORLD    P7,A            ; BUT LEAVE EEN ACTIVE
0111 F8          423        MOV     A,R0            ; RESTORE A
0112 83          424        RET
                 425
                 426
                 427        ;       ************************************************
                 428
                 429        ;               OUTADR - OUTPUT ADDRESSES TO ADDRESS LINES
                 430
                 431        ;               USES A-REG, R0
                 432
                 433        ;       ************************************************
                 434
                 435 OUTADR:
0113 B815        436        MOV     R0,#ADRHI       ; POINT TO HI ORDER ADDRESS REGISTER
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
0115 F0          437         MOV     A,@R0           ; GET HI ORDER ADDRESS
                 438
                 439         ;       HI ORDER ADDRESS OUTPUT (OUTPUT A12,A13 TO P26,P27)
                 440
0116 E7          441         RL      A               ; MOVE A12,A13 TO BITS 6,7
0117 E7          442         RL      A
0118 53C0        443         ANL     A,#0C0H         ; MAKE SURE ONLY BITS 6,7 SET
011A 3A          444         OUTL    P2,A            ; AND OUTPUT TO PORT 2
011B 230B        445         MOV     A,#EENACT       ; ACTIVATE EEN'
011D 9F          446         ANLD    P7,A
                 447
                 448 OUTA20:
011E B814        449         MOV     R0,#ADRLO       ; POINT TO LOW ORDER ADDRESS REGISTER
0120 F0          450         MOV     A,@R0           ; GET LOW ORDER ADDRESS
0121 3C          451         MOVD    P4,A            ; OUTPUT LOW ORDER ADDRESS
0122 47          452         SWAP    A
0123 3D          453         MOVD    P5,A
0124 18          454         INC     R0              ; POINT TO HI ORDER ADDRESS
0125 F0          455         MOV     A,@R0           ; GET HI ORDER ADDRESS
0126 3E          456         MOVD    P6,A            ; AND OUTPUT IT
0127 83          457         RET                     ; BACK TO CALLING PROGRAM
                 458
                 459
                 460
                 461         ;       ************************************************
                 462
                 463         ;               REQACC - REQUEST ACCESS TO BUS AND
                 464         ;                        SIEZE BUS
                 465
                 466         ;               USES:   A-REG
                 467
                 468         ;       ************************************************
                 469
                 470 REQACC:
0128 362F        471         JT0     GOTACC          ; IF ALREADY GOT ACCESS SKIP REQUEST
012A 2307        472         MOV     A,#REQACT       ; REQUEST ACCESS
012C 9F          473         ANLD    P7,A
012D 262D        474 WAITAC: JNT0    WAITAC          ; WAIT FOR ACCESS
                 475 GOTACC:
012F 83          476         RET                     ; BACK TO CALLER
                 477
                 478
                 479         ;       ************************************************
                 480
                 481         ;               RELEASE - RELEASE BUS
                 482
                 483         ;               USES:   A-REG
                 484
                 485         ;       ************************************************
                 486
                 487 RELEAS:
0130 23FF        488         MOV     A,#0FFH         ; WRITE 1'S TO ALL ADDR AND DATA LINES
0132 39          489         OUTL    P1,A
0133 3C          490         MOVD    P4,A
0134 3D          491         MOVD    P5,A
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
0135 3E          492         MOVD    P6,A
0136 8AC0        493         ORL     P2,#0C0H        ; SET P26,P27 = 1
                 494
0138 3F          495 RELRET: MOVD    P7,A
0139 83          496         RET                     ; RETURN TO CALLER
                 497
                 498
                 499         ;       ***********************************************
                 500
                 501         ;               REMADR - REMOVE ADDRESSES
                 502         ;                        (SET ADDRESS LINES = 1)
                 503         ;                        ALSO DEACTIVATES EEN' (SETS TO 1)
                 504
                 505         ;               USES:   A-REG
                 506
                 507         ;       ***********************************************
                 508
                 509 REMADR:
013A 23FF        510         MOV     A,#0FFH                 ; SET A = ALL 1'S
013C 8C          511         ORLD    P4,A                    ; OUTPUT TO A8-A11
013D 8D          512         ORLD    P5,A
013E 8E          513         ORLD    P6,A
013F 8AC0        514         ORL     P2,#0C0H
                 515
0141 2304        516 REMRET: MOV     A,#EENINA               ; DE-ACTIVATE 2816'S
0143 8F          517         ORLD    P7,A
0144 83          518         RET
                 519
                 520 $EJECT
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
                  521
                  522
                  523       ;       ********************************************
                  524
                  525       ;               WRITE - WRITE TO 2816
                  526
                  527       ;               USES:   R0,A
                  528
                  529       ;       ********************************************
                  530
                  531 WRITE:
0145 3407         532       CALL    READ            ; READ DATA FIRST
0147 37           533       CPL     A               ; NEGATE DATA
0148 A8           534       MOV     R0,A            ; SAVE COMPLEMENTED DATA
0149 17           535       INC     A               ; NEGATE A
014A 6D           536       ADD     A,R5            ; A = (DATA TO WRITE - DATA READ)
014B C658         537       JZ      WREND           ; IF SAME DATA, DON'T BOTHER WRITING
014D F8           538       MOV     A,R0            ; RETRIEVE NEGATED DATA FROM READ CYCLE
014E 5D           539       ANL     A,R5            ; ANY 0'S TO BE PROGRAMMED TO 1'S?
014F C655         540       JZ      WRCYCL          ; NO -> SKIP ERASE BYTE ROUTINE
0151 23FF         541       MOV     A,#0FFH         ; SET DATA TO WRITE = 0FFH
0153 3459         542       CALL    WECYCL          ; ERASE THE BYTE
                  543 WRCYCL:
0155 FD           544       MOV     A,R5            ; WRITE DATA NOW
0156 3459         545       CALL    WECYCL
0158 83           546 WREND:  RET                 ; RETURN BACK TO CALLING PROGRAM
                  547
                  548
                  549       ;       ********************************************
                  550
                  551       ;               WECYCL - WRITE DATA IN A-REG TO 2816
                  552
                  553       ;               ASSUMES:        EE ENABLED, ADDRESSES OUTPUT
                  554
                  555       ;               USES:   R0
                  556
                  557       ;       ********************************************
                  558 WECYCL:
0159 39           559       OUTL    P1,A            ; OUTPUT DATA TO WRITE
015A 230E         560       MOV     A,#VPPACT       ; ACTIVATE VPP
015C 9F           561       ANLD    P7,A            ; ACTIVATE VPP
015D B81D         562       MOV     R0,#EESTAT      ; LET R0 POINT TO EE STATUS REGISTER
015F F0           563       MOV     A,@R0           ; GET EE STATUS
0160 537F         564       ANL     A,#INITWR       ; CLEAR WRITE COMPLETE BIT
0162 A0           565       MOV     @R0,A           ; RESTORE STATUS REGISTER
0163 D5           566       SEL     RB1             ; CHOOSE RB1 TO GET INITIAL TIMER VALUE
0164 85           567       CLR     F0              ; SET F0 TO INDICATE VPP IS ON
0165 95           568       CPL     F0
0166 FE           569       MOV     A,R6            ; START TIMER
0167 62           570       MOV     T,A
0168 55           571       STRT    T
0169 25           572       EN      TCNTI           ; ENABLE TIMER/COUNTER INTERRUPTS
                  573 WRWAIT:
016A 3483         574       CALL    CKDBB           ; CHECK FOR ABORT COMMAND IN DBB
016C FD           575       MOV     A,R5            ; GET EE STATUS
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT

016D F271        576        JB7     WRENDC          ; SKIP RECHECK IF END OF CYCLE
016F 246A        577        JMP     WRWAIT
                 578 WRENDC:
0171 3474        579        CALL    SHUT            ; END OF WRITE CYCLE - SHUT DOWN 2816
0173 93          580        RETR                    ; RETURN BACK TO USER PROGRAM
                 581
                 582
                 583        ;       SHUT - SHUT OFF VPP, WAIT VPP FALL AND DEACTIVATE DATA LINES
                 584
                 585 SHUT:
0174 23F1        586        MOV     A,#VPPINA       ; DE-ACTIVATE VPP
0176 8F          587        ORLD    P7,A
0177 230E        588        MOV     A,#VPPFALL      ; WAIT WHILE VPP FALLS
0179 85          589        CLR     F0              ; CLEAR F0 TO INDICATE VPP IS OFF
017A 347F        590        CALL    DELAY
017C 89FF        591        ORL     P1,#0FFH        ; DEACTIVATE DATA LINES
017E 83          592        RET
                 593
                 594
                 595        ;       DELAY - SUBROUTINE TO DELAY
                 596
                 597        ;       PASS:   A-REG = COUNT (7.5 USEC/LOOP)
                 598
                 599 DELAY:
017F 07          600        DEC     A               ; DECREMENT A-REG
0180 967F        601        JNZ     DELAY           ; LOOP IF NOT ZERO
0182 83          602        RET                     ; BACK TO CALLER
                 603
                 604
                 605        ;       CKDBB - CHECK FOR ABORT COMMAND IN DBB
                 606
                 607        ;       USES:   A-REG
                 608
                 609 CKDBB:
0183 D687        610        JNIBF   CKDEX           ; IF INPUT BUFFER NOT FULL THEN EXIT
0185 7688        611        JF1     CKD10           ; IF COMMAND WAITING, CHECK IT
                 612 CKDEX:
0187 83          613        RET                     ; RETURN TO USER
                 614 CKD10:
0188 22          615        IN      A,DBB           ; GET COMMAND
0189 A5          616        CLR     F1              ; CLEAR COMMAND/DATA FLAG
018A 03F6        617        ADD     A,#-ABORTC      ; IS IT AN ABORT COMMAND?
018C 9690        618        JNZ     ILLCD1          , NO - ILLEGAL COMMAND
018E 6410        619        JMP     ABORT           ; GO ABORT
                 620 ILLCD1:
0190 23AB        621        MOV     A,#0ABH         ; OUTPUT ILLEGAL COMMAND MESSAGE
0192 02          622        OUT     DBB,A
0193 8A10        623        ORL     P2,#OBFACT      ; ALLOW OBF HOST INTERRRUPTS
0195 83          624        RET
                 625
                 626 $EJECT
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
                  627
                  628       ;     ********************************************
                  629
                  630       ;               GENERAL SUBROUTINES
                  631
                  632       ;     ********************************************
                  633
                  634
                  635
                  636       ;     INCADR - INCREMENT ADDRESS REGISTER
                  637
                  638       ;     USES:   RB0 - ANY REGISTER BANK
                  639       ;             A-REG
                  640
                  641 INCADR:
0196 B814         642         MOV   R0,#ADRLO       ; POINT TO LOW ADDRESS REGISTER
0198 10           643         INC   @R0             ; INCREMENT LOW ORDER ADDRESS
0199 F0           644         MOV   A,@R0           ; GET LOW ORDER ADDRESS
019A 969E         645         JNZ   INCEX           ; IF NONZERO, EXIT
019C 18           646         INC   R0              ; OVERFLOW - POINT TO HI ORDER ADDRESS REGISTER
019D 10           647         INC   @R0             ; INCREMENT HI ORDER ADDRESS
019E 83           648 INCEX:  RET                   ; RETURN TO USER PROGRAM
                  649
                  650
                  651
                  652       ;     DECCNT - DECREMENT COUNT REGISTER
                  653
                  654       ;     RETURNS:        A-REG = 0 IF COUNT = 0
                  655
                  656       ;     USES:           R0, A-REG
                  657
                  658 DECCNT:
019F B806         659         MOV   R0,#CNTLO       ; POINT TO LOW ORDER COUNT REGISTER
01A1 F0           660         MOV   A,@R0           ; GET COUNT
01A2 07           661         DEC   A               ; DECREMENT LOW ORDER COUNT
01A3 A0           662         MOV   @R0,A           ; RESTORE COUNT
01A4 C6AF         663         JZ    ZTEST           ; IF ALREADY ZERO, TEST FOR 0 COUNT
01A6 17           664         INC   A               ; SEE IF WE NEED TO BORROW
01A7 96AE         665         JNZ   DECEX           ; NO - JUST EXIT
01A9 18           666         INC   R0              ; YES - POINT TO HI ORDER COUNT REGISTER
01AA F0           667         MOV   A,@R0           ; GET HI ORDER COUNT VALUE
01AB 07           668         DEC   A               ; DECREMENT IT
01AC A0           669         MOV   @R0,A           ; RESTORE COUNT VALUE
01AD 17           670         INC   A               ; SET A = NONZERO TO INDICATE NOT 0 COUNT
01AE 83           671 DECEX:  RET                   ; RETURN TO USER
                  672 ZTEST:
01AF 18           673         INC   R0              ; POINT TO HI ORDER COUNT REG TO SEE IF IT'S 0
01B0 40           674         ORL   A,@R0           ; OR IN HI ORDER COUNT
01B1 24AE         675         JMP   DECEX           ; A = 0 IF HI OR LOW = 0 SO EXIT
                  676
                  677
                  678 $EJECT
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT

                679
                680         ;       ************************************************
                681
                682         ;                   COMMAND   DRIVER
                683
                684         ;       ************************************************
                685
                686
0200            687         ORG     200H            ; START COMMANDS ON PAGE 2
                688
                689 FIRSTI:
                690
                691         ;       ILLEGAL COMMAND
                692
                693 IC:
0200 042E       694         JMP     ILLCMD          ; ILLEGAL COMMAND
                695
                696
                697         ;       READ COMMAND
                698
                699 READC:
0202 3428       700         CALL    REQACC          ; REQUEST ACCESS TO LOCAL BUS
0204 3407       701         CALL    READ            ; READ FROM 2816
                702 RDCPL:
0206 540A       703         CALL    OUTPUT          ; OUTPUT DATA TO DBB
0208 0423       704 CMDEX:  JMP     CMDCPL          ; COMMAND EXITS HERE
                705
                706
                707         ;       OUTPUT - OUTPUT DATA IN A-REG TO DBB
                708
                709 OUTPUT:
020A 8A10       710         ORL     P2,#OBFACT      ; ACTIVATE OBF INTERRUPT
020C 02         711         OUT     DBB,A           ; OUTPUT DATA TO HOST
020D 860D       712 WAITOU: JOBF    WAITOU          ; WAIT UNTIL OUTPUTPUT BUFR NOT FULL
020F 83         713         RET
                714
                715
                716         ;       WRITE COMMAND
                717
                718 WRITEC:
0210 3428       719         CALL    REQACC          ; REQUEST ACCESS TO LOCAL BUS
0212 3445       720         CALL    WRITE           ; WRITE TO 2816
0214 0423       721         JMP     CMDCPL          ; COMMAND COMPLETE
                722
                723
                724         ;       CHIP ERASE COMMAND
                725
                726 CERASE:
0216 3428       727         CALL    REQACC          ; REQUEST ACCESS TO BUS
0218 B815       728         MOV     R0,#ADRHI       ; POINT TO HI ORDER ADDRESS
                729
                730         ;       OUTPUT A12,A13 TO P26,P27
                731
021A F0         732         MOV     A,@R0           ; GET HI ORDER ADDRESS
021B E7         733         RL      A               ; MOVE TO BITS 6,7
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT

021C E7          734        RL      A
021D 53C0        735        ANL     A,#0C0H         ; MAKE SURE BITS 6,7 THE ONLY ONES SET
021F 3A          736        OUTL    P2,A            ; OUTPUT HI ORDER ADDRESS BITS
0220 F0          737        MOV     A,@R0           ; GET HI ORDER ADDRESS
0221 997F        738        ANL     P1,#7FH         ; SET D7/CERASE' = 0 (DATA LINE)
0223 230B        739        MOV     A,#EENACT       ; ACTIVATE EEN' TO LATCH CERASE' BIT
0225 9F          740        ANLD    P7,A
0226 F0          741        MOV     A,@R0           ; GET HI ORDER ADDRESS AGAIN
0227 3E          742        MOVD    P6,A            ; OUTPUT ADDRESS BITS A8-A11
0228 23FF        743        MOV     A,#0FFH         ; SET DATA TO WRITE = 0FFH FOR CHIP ERASE
022A 3459        744        CALL    WECYCL          ; ERASE THE SELECTED CHIP
022C 5430        745        CALL    REMCE           ; REMOVE CHIP ERASE
022E 0423        746        JMP     CMDCPL          ; END OF COMMAND
                 747
                 748
                 749        ;       REMCE - REMOVE CHIP ERASE (OE'=12V)
                 750
                 751        ;       USES A-REG
                 752
                 753 REMCE:
0230 89FF        754        ORL     P1,#0FFH        ; SET DATA LINES = 0FFH
0232 2304        755        MOV     A,#EENINA       ; DE-ACTIVATE EEN
0234 8F          756        ORLD    P7,A
0235 230B        757        MOV     A,#EENACT       ; D7/CERASE' = 1 (SINCE DATA WE WRITE = FFH)
                 758                                ; JUST PULSE EEN TO LATCH INACTIVE CERASE'
0237 9F          759        ANLD    P7,A            ; PULSE EEN' TO LATCH INACTIVE CERASE'
0238 83          760        RET
                 761
                 762
                 763        ;       BLOCK ERASE
                 764
                 765 BLOCKE:
0239 3428        766        CALL    REQACC          ; REQUEST ACCESS TO BUS
023B BDFF        767        MOV     R5,#0FFH        ; SE DATA TO WRITE = 0FFH
                 768 BL10:
023D 3445        769        CALL    WRITE           ; WRITE ERASED BYTE
023F 343A        770        CALL    REMADR          ; DE-ACTIVATE ADDRESS LINES
0241 3496        771        CALL    INCADR          ; INCREMENT EE ADDRESS
0243 EE3D        772        DJNZ    R6,BL10         ; KEEP LOOPING IF NOT DONE
0245 0423        773        JMP     CMDCPL          ; WHEN DONE EXIT
                 774
                 775
                 776        ;       MULTIPLE WRITE
                 777
                 778 MULTWR:
0247 BC00        779        MOV     R4,#0           ; ZERO BUFFER COUNT
0249 D5          780        SEL     RB1             ; SELECT ALTERNATE REGISTER SET
024A BF20        781        MOV     R7,#BUFST       ; POINT TO START OF BUFFER
                 782 MWGET:
024C D5          783        SEL     RB1             ; MAKE SURE WE'RE POINTING TO RB1
024D FB          784        MOV     A,R3            ; GET DBB STATUS
024E 53E0        785        ANL     A,#0E0H         ; ZERO TYPE & BYTE STATUS
0250 431D        786        ORL     A,#11101B       ; SET TYPE=7, 3 BYTES TO GET (SPECIAL TYPE)
0252 AB          787        MOV     R3,A            ; RESTORE DBB STATUS
0253 149F        788        CALL    GETDAT          ; GET 3 BYTES FROM HOST
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE        SOURCE STATEMENT
0255 FF          789          MOV     A,R7            ; GET DATA DESTINATION POINTER
0256 C5          790          SEL     RB0
0257 1C          791          INC     R4              ; BUMP BUFFER COUNTER
0258 03C2        792          ADD     A,#(-MWBEND)    ; CHECK FOR END OF BUFFER
025A C65F        793          JZ      MWDUMP          ; IF END GO DUMP BUFFER
025C EE4C        794          DJNZ    R6,MWGET        ; IF NOT DONE THEN GET MORE
025E 1E          795          INC     R6              ; INCREMENT BECAUSE WE'LL DEC NEXT CYCLE
                 796 MWDUMP:
025F CE          797          DEC     R6              ; DECREMENT BECAUSE MAY NOT HAVE DONE SO
0260 B920        798          MOV     R1,#BUFST       ; POINT TO START OF BUFFER
                 799 MWDMP1:
0262 B815        800          MOV     R0,#ADRHI       ; POINT TO HI ORDER ADDRESS
0264 3428        801          CALL    REQACC          ; REQUEST ACCESS TO BUS
0266 F1          802          MOV     A,@R1           ; GET HI ADDRESS AGAIN
0267 A0          803          MOV     @R0,A           ; SAVE IN HI ADDRESS REGISTER
0268 C8          804          DEC     R0              ; POINT TO LOW ADDRESS REG
0269 19          805          INC     R1              ; BUMP BUFFER POINTER
026A F1          806          MOV     A,@R1           ; GET LOW ORDER ADDRESS
026B A0          807          MOV     @R0,A           ; AND SAVE IN LOW ORDER ADDRESS REGISTER
026C 18          808          INC     R0              ; POINT TO HI ADDRESS
026D 19          809          INC     R1              ; BUMP BUFFER POINTER
026E F1          810          MOV     A,@R1           ; GET DATA TO WRITE
026F 19          811          INC     R1              ; BUMP BUFFER POINTER
0270 AD          812          MOV     R5,A            ; SAVE IN DATA WRITE REGISTER
0271 3445        813          CALL    WRITE           ; WRITE TO 2816
0273 343A        814          CALL    REMADR          ; DE-ACTIVATE ADDRESS LINES
0275 EC62        815          DJNZ    R4,MWDMP1       ; DECREMENT COUNT - KEEP WRITING IF NOT DONE
0277 FE          816          MOV     A,R6            ; PICK UP COUNT TO SEE IF WE'VE DONE ALL BYTES
0278 C67C        817          JZ      MWEX            ; IF DONE THEN EXIT
027A 4447        818          JMP     MULTWR          ; IF NOT DONE THEN GET ANOTHER BLOCK
027C 0423        819 MWEX:    JMP     CMDCPL          ; EXIT TO COMMAND COMPLETE ROUTINE
                 820
                 821
                 822          ;       SERIES WRITE
                 823
                 824 SERWRT:
027E BC00        825          MOV     R4,#0           ; ZERO BUFFER COUNTER
0280 D5          826          SEL     RB1             ; POINT TO ALTERNATE REGISTER
0281 BF20        827          MOV     R7,#BUFST       ; POINT TO START OF BUFFER IN DESTINATION REG
                 828 SWGET:
0283 D5          829          SEL     RB1             ; MAKE SURE IN REGISTER BANK 1
0284 FB          830          MOV     A,R3            ; GET DBB STATUS
0285 53E0        831          ANL     A,#0E0H         ; REMOVE TYPE AND BYTE BITS
0287 431F        832          ORL     A,#11111B       ; OR INT TYPE=4, 1 TYPE TO GET
0289 AB          833          MOV     R3,A            ; RESTORE DBB STATUS
028A 149F        834          CALL    GETDAT          ; GET 1 BYTE FROM HOST
028C FF          835          MOV     A,R7            ; GET BUFFER POINTER
028D C5          836          SEL     RB0             ; POINT TO NORMAL REGISTER SET
028E 1C          837          INC     R4              ; INCREMENT BUFFER COUNTER
028F 03C0        838          ADD     A,#-SWBEND      ; CHECK FOR END OF BUFFER
0291 C699        839          JZ      SWDMP0          ; IF END OF BUFFER THEN GO DUMP DATA TO EEPROM
0293 349F        840          CALL    DECCNT          ; DECREMENT COUNT
0295 9683        841          JNZ     SWGET           ; IF COUNT NOT ZERO THEN GET ANOTHER BYTE
0297 449B        842          JMP     SWDUMP          ; SKIP OVER FIX, SINCE K-CODE MAKES IT NEC.
                 843 SWDMP0:
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT
0299 349F        844        CALL    DECCNT          ; SKIPPED DECREMENT ABOVE, SO FIX NOW
                 845 SWDUMP:
029B B920        846        MOV     R1,#BUFST       ; POINT TO START OF BUFFER
029D 3428        847        CALL    REQACC          ; REQUEST ACCESS TO INTERNAL BUSSES
                 848 SWDMP1:
029F F1          849        MOV     A,@R1           ; GET A BYTE OF DATA
02A0 AD          850        MOV     R5,A            ; SAVE DATA TO WRITE
02A1 3445        851        CALL    WRITE           ; WRITE TO 2816
02A3 343A        852        CALL    REMADR          ; RELEASE ADDRESS LINES
02A5 19          853        INC     R1              ; BUMP POINTER
02A6 3496        854        CALL    INCADR          ; INCREMENT EE ADDRESS
02A8 EC9F        855        DJNZ    R4,SWDMP1       ; DECREMENT COUNT - IF NOT ZERO, WRITE NXT BYTE
                 856
                 857        ;       DONE WRITING BUFFER - CHECK IF DONE WITH ALL BYTES
                 858
02AA FE          859        MOV     A,R6            ; CHECK IF COUNT BYTES ARE 0
02AB 4F          860        ORL     A,R7            ; OR IN HI AND LOW ORDER COUNT BYTES
02AC 967E        861        JNZ     SERWRT          ; IF NONZERO, GET ANOTHER BUFFER FULL OF DATA
02AE 0423        862        JMP     CMDCPL          ; IF DONE GO TO COMMAND COMPLETE ROUTINE
                 863
                 864
                 865        ;       SERIES READ COMMAND
                 866
                 867 SERRD:
02B0 3428        868        CALL    REQACC          ; REQUEST ACCESS TO LOCAL BUSSES
02B2 3407        869        CALL    READ            ; READ FROM 2816 MEMORY
02B4 540A        870        CALL    OUTPUT          ; OUTPUT SERIALLY OR TO DATA BUS BUFFER
02B6 343A        871        CALL    REMADR          ; RELEASE ADDRESS PINS
02B8 3496        872        CALL    INCADR          ; INCREMENT EE ADDRESS
02BA 349F        873        CALL    DECCNT          ; DECREMENT COUNT AND SET Z FLAG IF COUNT=0
02BC 96B0        874        JNZ     SERRD           ; READ AGAIN IF NOT DONE
02BE 0423        875        JMP     CMDCPL          ; GO TO COMMAND COMPLETE ROUTINE
                 876
                 877
                 878
02C0 4400        879 ICDRD:  JMP     IC              ; OFF-PAGE REFERENCE TO ILLEGAL COMMAND ROUTINE
                 880
                 881
0300             882        ORG     300H
                 883
                 884 FIRSTP3:
                 885
                 886        ;       ENABLE DIRECT WRITE COMMAND
                 887
                 888 ENDW:
0300 D5          889        SEL     RB1             ; SELECT ALTERNATE REGISTER SET
0301 FB          890        MOV     A,R3            ; GET DBB STATUS
0302 4340        891        ORL     A,#040H         ; SET DIRECT WRITE BIT
0304 AB          892        MOV     R3,A            ; RESTORE DBB STATUS
0305 85          893        CLR     F0              ; SET DIRECT WRITE POSSIBLE FLAG
0306 95          894        CPL     F0
0307 0423        895        JMP     CMDCPL          ; COMMAND COMPLETE
                 896
                 897
                 898        ;       DISABLE DIRECT WRITE COMMAND
```

APPENDIX D: 8298 E^2PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT
                  899
                  900 DISDW:
0309 D5           901         SEL     RB1             ; POINT TO ALTERNATE REGISTER SET
030A FB           902         MOV     A,R3            ; GET DBB STATUS
030B 53BF         903         ANL     A,#NOT 40H      ; CLEAR D/W BIT AS WELL AS EXTRANEOUS BIT
030D AB           904         MOV     R3,A            ; RESTORE DBB STATUS
030E 0423         905         JMP     CMDCPL          ; GO TO COMMAND COMPLETE ROUTINE
                  906
                  907
                  908         ;       ABORT - DO A SOFTWARE RESET
                  909
                  910 ABORT:
0310 3F           911         MOVD    P7,A            ; SEE IF VPP ON
0311 1215         912         JB0     AB10            ; IF YES, TURN OFF VPP
0313 3474         913         CALL    SHUT
                  914 AB10:
0315 5430         915         CALL    REMCE           ; REMOVE CHIP ERASE IN CASE IT WAS ON
0317 0423         916         JMP     CMDCPL          ; END OF COMMAND
                  917
                  918
                  919         ;       COMMANDS TO BE USED AFTER AN ABORT
                  920
                  921         ;       READ LOW ADDRESS COMMAND
                  922
                  923 READAL:
0319 B814         924         MOV     R0,#ADRLO       ; POINT TO LOW ORDER ADDRESS REG.
031B F0           925         MOV     A,@R0           ; READ IT INTO ACC.
031C 07           926         DEC     A               ; DECREMENT TO GIVE CORRECT VALUE
031D 4406         927         JMP     RDCPL
                  928
                  929         ;       READ HIGH ADDRESS COMMAND
                  930
                  931 READAH:
031F B815         932         MOV     R0,#ADRHI       ; POINT TO HIGH ORDER ADDRESS REG.
0321 F0           933         MOV     A,@R0
0322 4406         934         JMP     RDCPL
                  935
                  936         ;       READ WRITE DATA COMMAND
                  937
                  938 READWR:
0324 FD           939         MOV     A,R5
0325 4406         940         JMP     RDCPL
                  941
                  942 $EJECT
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT
                  943
                  944       ;       ************************************************
                  945
                  946       ;               PAGE 3 DATA TABLES
                  947
                  948       ;       ************************************************
                  949
0365              950       ORG     365H
                  951
                  952
                  953
                  954       ;       INSTRUCTION TABLE
                  955
                  956       ;       FORMAT:
                  957       ;
                  958       ;       1ST BYTE = COMMAND DESCRIPTOR:
                  959       ;
                  960       ;               BIT(S)          DESCRIPTION
                  961       ;               ------          -----------
                  962       ;               0-1             STARTING BYTE # (0-3) OF DATA BYTES
                  963       ;                               TO GET FOR THIS COMMAND
                  964       ;               2-4             TYPE (FROM DATA DEST TABLE) OF DATA
                  965       ;                               BYTES TO GET
                  966       ;                5              NO DATA BYTES TO RECEIVE FOR THIS
                  967       ;                               COMMAND (=1)
                  968       ;                6              PAGE 3 COMMAND
                  969       ;                7              COMMAND CAN BE EXECUTED UNDER SERIAL
                  970       ;                               I/O MODE
                  971       ;
                  972       ;       2ND BYTE = LOW ORDER STARTING ADDRESS OF COMMAND
                  973
                  974       ;       COMMENT FORMAT: CMND #,COMMAND, TYPE OF DATA, # BYTES TO GET
                  975
                  976
                  977 INSTBL:
0365 20           978       DB      20H,LOW IC              ; SUB COMMAND - WE SHOULDN'T GET HERE
0366 00
0367 83           979       DB      10000011B,LOW READC     ; 1 = READ, TYPE=0, BYTE=1
0368 02
0369 86           980       DB      10000110B, LOW WRITEC   ; 2 = WRITE, TYPE=1,BYTES=2
036A 10
036B 20           981       DB      20H,LOW IC              ; 3 = IC
036C 00
                  982
                  983       ;       SUB COMMANDS
                  984
                  985 SUBTBL:
036D 20           986       DB      20H,LOW IC              ; 0 = ILLEGAL COMMAND
036E 00
036F 20           987       DB      20H,LOW IC              ; 1 = ILLEGAL COMMAND
0370 00
0371 20           988       DB      20H,LOW IC              ; 2 = IC
0372 00
0373 20           989       DB      20H,LOW IC              ; 3 = IC
0374 00
```

APPENDIX D: 8298 E^{2}PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE      SOURCE STATEMENT

0375 20          990       DB      20H,LOW IC              ; 4 = IC
0376 00
0377 09          991       DB      00001001B,LOW BLOCKE    ; 5 = BLOCK ERASE, TYPE=2, BYTES=3
0378 39
0379 9B          992       DB      10011011B,LOW CERASE    ; 6 = CHIP ERASE, TYPE=6, BYTES=1
037A 16
037B 17          993       DB      00010111B,LOW CMDEX     ; 7 = INIT WRITE TIME, TYPE=5,BYTES=1
037C 08
037D 20          994       DB      20H,LOW IC              ; 8 = IC
037E 00
037F 20          995       DB      20H,LOW IC              ; 9 = IC
0380 00
0381 60          996       DB      01100000B,LOW ABORT     ; A = ABORT, TYPE=0, BYTES=0
0382 10
0383 20          997       DB      20H,LOW IC              ; B = IC
0384 00
0385 8B          998       DB      10001011B,LOW MULTWR    ; C = MULTIPLE WRITE, TYPE=2,BYTE=1
0386 47
                 999
                 1000      REPT    19D
                 1001      DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
                 1002      ENDM
0387 20          1003+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0388 00
0389 20          1004+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
038A 00
038B 20          1005+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
038C 00
038D 20          1006+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
038E 00
038F 20          1007+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0390 00
0391 20          1008+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0392 00
0393 20          1009+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0394 00
0395 20          1010+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0396 00
0397 20          1011+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
0398 00
0399 20          1012+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
039A 00
039B 20          1013+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
039C 00
039D 20          1014+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
039E 00
039F 20          1015+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03A0 00
03A1 20          1016+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03A2 00
03A3 20          1017+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03A4 00
03A5 20          1018+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03A6 00
03A7 20          1019+     DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
```

APPENDIX D: 8298 E²PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT
03A8 00
03A9 20         1020+       DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03AA 00
03AB 20         1021+       DB      20H,LOW IC              ; D-1F = ILLEGAL COMMAND
03AC 00
                1022
03AD 20         1023        DB      20H,LOW IC              ; 20 = IC
03AE 00
03AF 20         1024        DB      20H,LOW IC              ; 21 = IC
03B0 00
03B1 60         1025        DB      01100000B,LOW ENDW      ; 22 = ENABLE D/W
03B2 00
03B3 60         1026        DB      01100000B,LOW DISDW     ; 23 = DISABLE D/W
03B4 09
03B5 8C         1027        DB      10001100B,LOW SERRD     ; 24 = SERIES READ, TYPE=3, BYTES=4
03B6 B0
03B7 8C         1028        DB      10001100B,LOW SERWRT    ; 25 = SERIES WRITE,TYPE=3, BYTES=4
03B8 7E
03B9 20         1029        DB      20H,LOW IC              ; 26 = IC
03BA 00
03BB 20         1030        DB      20H,LOW IC              ; 27 = IC
03BC 00
03BD 60         1031        DB      01100000B,LOW READAL    ; 28 = READ LOW ADDR, NO DATA
03BE 19
03BF 60         1032        DB      01100000B,LOW READAH    ; 29 = READ HI  ADDR, NO DATA
03C0 1F
03C1 60         1033        DB      01100000B,LOW READWR    ; 2A = READ WRITE DATA, NO DATA TO RCV
03C2 24
                1034
                1035        REPT 5D
                1036        DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
                1037        ENDM                            ; 30-3F DETECTED EARLIER
03C3 20         1038+       DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
03C4 00
03C5 20         1039+       DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
03C6 00
03C7 20         1040+       DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
03C8 00
03C9 20         1041+       DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
03CA 00
03CB 20         1042+       DB      20H,LOW IC              ; 2B-3F = ILLEGAL COMMAND
03CC 00
                1043
                1044
                1045        ;       DATA DESTINATION TABLE
                1046
                1047        ;       FORMAT:
                1048        ;               DB      ADR OF BYTE 3, BYTE 2, BYTE 1, BYTE 0 ;TYPE=N
                1049
                1050 DESTBL:
03CD 00         1051        DB      0,0,0,ADRLO             ; TYPE 0
03CE 00
03CF 00
03D0 14
03D1 00         1052        DB      0,ADRHI,ADRLO,WRTDAT    ; TYPE 1
```

APPENDIX D: 8298 E^2PROM CONTROLLER FIRMWARE LISTING

```
LOC  OBJ        LINE       SOURCE STATEMENT

03D2 15
03D3 14
03D4 05
03D5 00         1053       DB      0,ADRHI,ADRLO,CNTLO     ; TYPE 2
03D6 15
03D7 14
03D8 06
03D9 15         1054       DB      ADRHI,ADRLO,CNTHI,CNTLO ; TYPE 3
03DA 14
03DB 07
03DC 06
03DD 00         1055       DB      0,0,0,ADRLO             ; TYPE 4
03DE 00
03DF 00
03E0 14
03E1 00         1056       DB      0,0,0,INTIME            ; TYPE 5
03E2 00
03E3 00
03E4 1E
03E5 00         1057       DB      0,0,0,ADRHI             ; TYPE 6
03E6 00
03E7 00
03E8 15
                1058
                1059       END

USER SYMBOLS
ABIO   0315  ABORT  0310  ABORTC 000A  ADRHI  0015  ADRLO  0014  ASAVE  001C  BL10   023D  BLOCKE 0239
BUFCNT 0004  BUFST  0020  CERASE 0216  CKD10  0188  CKDBB  0183  CKDEX  0187  CMDCPL 0023  CMDEX  0208
CNTHI  0007  CNTLO  0006  COMFB  001A  DATDES 001F  DATRCV 00C7  DBBI   0033  DBBI07 003F  DBSTAT 001B
DECCNT 019F  DECEX  01AE  DELAY  017F  DESTBL 03CD  DISDW  0309  DW05   00E1  DW10   00F7  DWCMD  00DD
DWE    001F  DWNE   0020  EENACT 000B  EENINA 0004  EESTAT 001D  ENDW   0300  EXEC   00C3  EXIT   0072
FIRSTI 0200  FIRSTP 0300  GETCMD 0015  GETDAT 009F  GETDBB 00B3  GOTACC 012F  IBFACT FF20  IBFINA 00DF
IC     0200  ICDRD  02C0  ILLCD1 0190  ILLCDO 006D  ILLCMD 002E  INCADR 0196  INCEX  019E  ININT  0016
INIT   0009  INITWR 007F  INSTBL 0365  INTCNT 0003  INTIME 001E  LOOKUP 0056  MAYBIC 004B  MULTWR 0247
MWBEND 003E  MWDMP1 0262  MWDUMP 025F  MWEX   027C  MWGET  024C  NOTICD 004D  OBFACT 0010  OBFINA FFEF
OUTA20 011E  OUTADR 0113  OUTPUT 020A  P3INST 0098  PC03   0080  PC05   008D  PC10   009A  PCMND  007D
RDACT  0009  RDCPL  0206  RDINA  0002  READ   0107  READAH 031F  READAL 0319  READC  0202  READWR 0324
RELEAS 0130  RELRET 0138  REMADR 013A  REMCE  0230  REMRET 0141  REQACC 0128  REQACT 0007  RESET  0000
SAVEHI 00B9  SCR0   0000  SCR0P  0018  SCR1   0001  SCR1P  0019  SCR2   0002  SERRD  02B0  SERWRT 027E
SETGET 00A1  SHUT   0174  SUBCMD 0075  SUBTBL 036D  SWBEND 0040  SWDMP0 0299  SWDMP1 029F  SWDUMP 029B
SWGET  0283  TIME   FF83  TIMER  00FC  VPPACT 000E  VPPFAL 000E  VPPINA FFF1  WAITAC 012D  WAITB  00B5
WAITOU 020D  WECYCL 0159  WIPSTS 0007  WRCYCL 0155  WREND  0158  WRENDC 0171  WRITE  0145  WRITEC 0210
XRTDAT 0005  WRWAIT 016A  ZTEST  01AF
ASSEMBLY COMPLETE,  NO ERRORS
```